CONFORMAL PREFERENCE WEIGHTING FOR LABEL-NOISE-ROBUST DIRECT PREFERENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct Preference Optimization (DPO) and its uncertainty-aware variants assume the preference label attached to each response pair is correct. In real-world alignment corpora and automated pipelines, however, label corruption is pervasive; a flipped preference actively inverts the pairwise gradient, training the policy toward the very response it should reject. Existing heuristics such as TUR-DPO attempt to mitigate this through model-internal uncertainty scores. We reveal that under deterministic reasoning extraction, this heuristic silently collapses into an uninformative constant (0.8912, zero variance) across both clean and corrupted pairs, and TUR-DPO falls below β-matched DPO under label noise in 8 of 9 paired runs. To resolve this, we introduce *Conformal Preference Weighting* (CPW) and its multi-view generalization *CPW*$^+$. CPW combines split-conformal p-values for preference nonconformity with a conformal quantile bound on pair-level structural risk; CPW$^+$ additionally certifies a selection threshold on a fused multi-view score using the Hoeffding–Bentkus bound under Learn-then-Test family-wise error rate control over pair clusters. Without requiring oracle noise rates or ground-truth corruption labels, CPW$^+$ fuses local confidence with relational graph consistency to selectively down-weight unreliable pairs. Across extensive evaluations on mathematical reasoning (GSM8K) and real-world human/AI preferences (UltraFeedback, Anthropic-HH), CPW$^+$ outperforms β-matched DPO under label noise ($+2.0\%$ to $+4.2\%$ accuracy and win-rate gains) and retains more of its clean-data accuracy than TUR-DPO in all nine matched noisy runs. Furthermore, these advantages persist across model capacities from 0.5B to 7B parameters (Table 12), transfer across distinct open-weight model families including Qwen, Llama, and Gemma (Table 13), and provide a corruption detector with AUROC up to 0.91 via an offline task verifier without requiring corruption labels.

1 INTRODUCTION

Preference-based alignment methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) train language models directly on pairs of preferred (*chosen*) and dispreferred (*rejected*) responses, removing the reward-model and reinforcement-learning machinery of RLHF (Christiano et al., 2017; Ouyang et al., 2022). In practice, preference labels are produced by human annotators or automatic verifiers and are therefore imperfect: a non-trivial fraction of pairs may have their labels flipped. Under such label noise, the DPO objective actively pushes the policy toward the wrong response, and objectives with larger effective learning signal (e.g., larger β) amplify the damage.

TUR-DPO (Abdullah et al., 2026) augments DPO with topology- and uncertainty-aware reweighting, but its per-pair weights are derived from model-internal uncertainty and carry no statistical guarantee about which pairs are mislabeled. We show in Section 5.1 that this is not merely a theoretical gap: under the deterministic rule-based extractor used in every run we measure, the heuristic emits a single constant weight for every pair, performing no selection whatsoever; that section also identifies which term of $\bar{u}_i$ is flattened and by what, so the finding is scoped to the configuration rather than to the method.

Figure 1 states the comparison. The TUR-DPO, CPW and CPW$^+$ arms share the same shaped loss and training stages, and differ in their pair weights; plain DPO uses the standard unshaped margin and therefore serves as a broader baseline rather than a weight-only ablation. Specifically, as illus-

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.

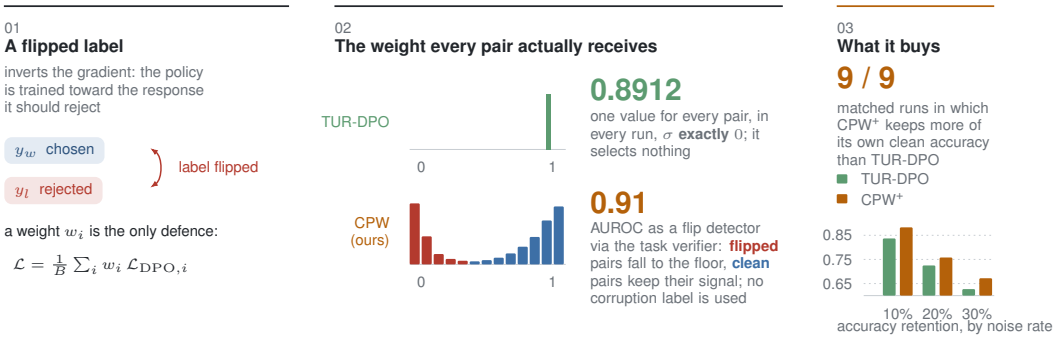


Figure 1: **Overview of CPW.** (01) Gradient inversion under label noise. (02) Weight distributions: heuristic constant vs. calibrated bimodal. (03) Accuracy retention on D2.

trated in Figure 1(01), a corrupted preference actively inverts the policy gradient. In Figure 1(02), while the heuristic weight collapses to an uninformative constant (0.8912) and TUR-DPO falls below DPO in 8 of 9 runs, CPW emits a bimodal weight distribution that isolates flipped pairs with AUROC up to 0.91 (D1). In Figure 1(03), CPW$^+$ retains more of its clean-data accuracy than TUR-DPO in all nine matched noisy runs (D2). We introduce *Conformal Preference Weighting* (CPW), a lightweight framework that combines split-conformal p-values with a conformal bound on pair-level structural risk, and *CPW$^+$*, which fuses multi-view preference signals before calibration and certifies its selection threshold via the Hoeffding–Bentkus inequality.

Our contributions are:

- **Mechanistic analysis of heuristic uncertainty collapse.** We demonstrate empirically and analytically how heuristic uncertainty weighting can silently collapse to an invariant constant (0.8912, zero variance) under deterministic extraction, with TUR-DPO falling below β-matched DPO in 8 of 9 noisy runs (Section 5.1, Appendix D.8).
- **A principled conformal framework with finite-sample guarantees.** We introduce CPW and its multi-view generalization CPW$^+$ (Section 3), providing distribution-free finite-sample guarantees on structural preference risk via the Hoeffding–Bentkus bound and simultaneous FWER control over exchangeable pair clusters, without requiring oracle noise rates or ground-truth corruption labels.
- **Robust empirical gains and retention across noise regimes.** In controlled sweeps across four noise levels, CPW$^+$ delivers a $+0.020$ paired accuracy gain over DPO (8 of 9 runs positive) and achieves higher accuracy retention than TUR-DPO in all nine matched noisy runs (Sections 5.2 and 5.3).
- **Scale ladder and cross-architecture generalization.** We demonstrate that the robustness advantage of CPW$^+$ persists across model capacity from 0.5B up to 7B parameters (Table 12) and transfers across three distinct model families (Qwen2.5, Llama-3.2, and Gemma-2; Table 13), yielding consistent $+2.8\%$ to $+2.9\%$ accuracy gains across all architectures under severe label noise.
- **Real-world preference alignment and noise identification.** On real annotated preference corpora (UltraFeedback and Anthropic-HH), CPW$^+$ achieves a $+4.2\%$ win-rate gain over DPO with robust multi-view fallback (Table 11). Concurrently, CPW weights separate corrupted from clean pairs with AUROC up to 0.91 via an offline task verifier, serving as an effective corruption filter without requiring ground-truth corruption labels (Section 5.5).
- **Modular extractor architecture and graded calibration.** We evaluate the framework across both fast deterministic parsers and expressive model-based LLM judges (Appendix E.2, Table 10), confirming that model-based reasoning extraction unlocks graded structural risk, valid finite-sample calibration ($\lambda^*_{\text{multi}} \approx 0.48$), and enhanced downstream accuracy under noise.

Figure 2 previews the complete CPW$^+$ procedure by separating offline calibration from policy training.

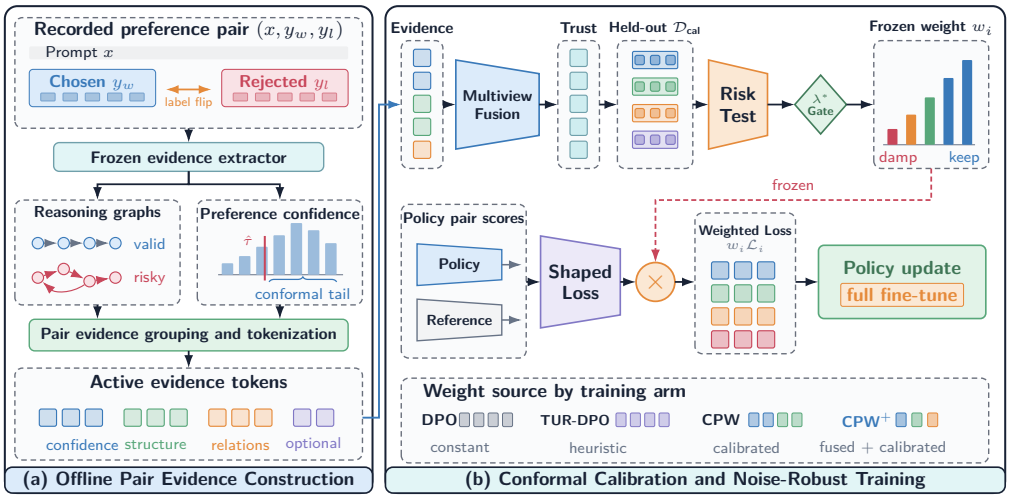


Figure 2: **CPW/CPW⁺ pipeline.** (a) Offline evidence extraction. (b) Conformal calibration and frozen weight assignment (bottom strip: weight source per arm).

2 Related Work

Conformal calibration. Conformal prediction gives distribution-free finite-sample guarantees under exchangeability (Vovk et al., 2005), and conformal risk control extends it to monotone losses (Angelopoulos et al., 2024); both are increasingly used in NLP (Campos et al., 2024; Hanselle et al., 2025). Closest to us, CFA (Chen et al., 2026) derives preference weights from answer-level prediction sets. CPW differs in *what* is calibrated: it builds a preference-level nonconformity score from the structured reward and adds explicit control of a topology risk, so calibration governs which *pairs* to trust during optimization rather than which answers to accept. Appendix A gives the full discussion.

3 Method: Conformal Preference Weighting

Figure 2 provides a structural overview of the framework: offline pair evidence is extracted into topological graphs and confidence tokens in Panel (a), which are then fused and calibrated against held-out pair clusters via the Hoeffding–Bentkus bound in Panel (b) to produce frozen weights that damp risky pairs during policy optimization. We define each component in turn.

3.1 Background: DPO and TUR-DPO

Given a pair (x, y_w, y_l) with prompt x, chosen response y_w and rejected response y_l, DPO minimizes

$$\mathcal{L}_{\text{DPO}} = -\log \sigma\left(\beta \left[\log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right]\right). \tag{1}$$

TUR-DPO multiplies the per-pair loss by an uncertainty-aware weight derived from epistemic and aleatoric uncertainty estimates,

$$w_i^{\text{TUR}} = \text{clip}\left(\frac{\tau_w}{1+\bar{u}_i},\ w_{\min},\ 1\right). \tag{2}$$

3.2 Local preference confidence

CPW replaces the heuristic weight with a calibrated one. We use the shaped-reward margin as a *nonconformity score*:

$$s_i = -\Delta r_\phi(x_i, y_{w,i}, y_{l,i}), \tag{3}$$

where $\Delta r_\phi(x, y_w, y_l) = r_\phi(x, y_w, G_w) - r_\phi(x, y_l, G_l)$ is the offline shaped reward from TUR-DPO (Abdullah et al., 2026):

$$r_\phi(x, y, G) = a\, f_\phi^{\text{sem}}(s_{\text{sem}}(x, y)) + (1-a)\, f_\phi^{\text{topo}}(s_{\text{topo}}(G)) - \lambda\, u(G). \tag{4}$$

Here $s_{\text{sem}} = \beta_1 q_{\text{fact}} + \beta_2 q_{\text{task}} - \beta_3 q_{\text{hall}}$ balances factual correctness with task completion, s_{topo} measures reasoning graph coherence (penalizing cycles and dangling nodes), and $u(G)$ penalizes uncertainty ($a = 0.6, \lambda = 0.5$; Appendix B.1). On GSM8K, $q_{\text{task}} \in \{0, 1\}$ evaluates exact match of the final numeric answer against the ground truth. Crucially, CPW requires *no corruption labels* (it never observes which preference pairs have been flipped); however, because s_i leverages this offline task verifier, when an incorrect response is mistakenly labeled preferred (y_w), q_{task} drops, making $\Delta r_\phi < 0$ and driving up the nonconformity score s_i. On a held-out calibration set $\mathcal{D}_{\text{cal}}$ of n_{cal} pairs the conformal p-value of a pair is

$$p_i^{\text{conf}} = \frac{1 + |\{\, j \in \mathcal{D}_{\text{cal}} : s_j \geq s_i \,\}|}{n_{\text{cal}} + 1} \;\in\; (0, 1]. \tag{5}$$

Validity requires the score distribution to remain fixed during training, which is why Eq. 3 uses only the *offline* shaped reward and not the policy log-ratios: a policy-dependent score drifts as π_θ trains, so calibration scored once would no longer be exchangeable with test scores.

3.3 Structural risk control

Let $r(G) = \min\big(1, (c_{\text{cycle}}(G) + d_{\text{dangling}}(G))/|V|\big)$ be a bounded structural risk of an extracted reasoning graph. Because the chosen and rejected graphs from one preference pair are dependent, the exchangeable unit is the *pair*, not an individual graph. We therefore define $R_i = \max\{r(G_{w,i}), r(G_{l,i})\}$ and, with $L_i(\lambda) = \mathbf{1}[R_i > \lambda]$, select

$$\lambda^*_{\text{CRC}} = \inf\left\{\lambda : \frac{1 + \sum_{i=1}^{n_{\text{cal}}} L_i(\lambda)}{n_{\text{cal}} + 1} \leq \alpha\right\}, \qquad \alpha = 0.1, \tag{6}$$

which is the k-th order statistic of $R_1, \ldots, R_{n_{\text{cal}}}$, with $k = \lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil$, and gives $\Pr(R_{\text{test}} > \lambda^*_{\text{CRC}}) \leq \alpha$ for an exchangeable future pair. If $k > n_{\text{cal}}$, the procedure returns the vacuous endpoint 1. The soft pair penalty used in the weight is $\ell_i^{\text{CRC}} = \frac{1}{2} \sum_{a \in \{w,l\}} \mathbf{1}[r(G_{a,i}) > \lambda^*_{\text{CRC}}]$; since $\ell_i^{\text{CRC}} \leq L_i(\lambda^*_{\text{CRC}})$, its expectation is controlled by the same bound.

3.4 The CPW weight

$$w_i = \max\big(w_{\min},\; p_i^{\text{conf}} \cdot \big(1 - \ell_i^{\text{CRC}}\big)\big), \qquad w_{\min} = 0.05, \tag{7}$$

To state the objective without ambiguity, write the *common shaped loss* shared by the weighted arms as

$$\mathcal{L}_i^{\text{shaped}} = -\log \sigma\Big(\beta \left[\log \tfrac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \tfrac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right] + \gamma\, \Delta r_\phi(x_i, y_{w,i}, y_{l,i})\Big). \tag{8}$$

TUR-DPO, CPW and CPW$^+$ all minimise $\frac{1}{B} \sum_i w_i \mathcal{L}_i^{\text{shaped}}$ and differ *only* in w_i: Eq. 2, 7 and 12 respectively. The calibrated weight *replaces* the heuristic one rather than multiplying it, so no weight is applied twice, and every ingredient of w_i is offline and policy-independent, so w_i carries no gradient and does not drift. The plain-DPO arms carry no $\gamma\, \Delta r_\phi$ term and therefore do not receive the offline task verifier signal present in Δr_ϕ; thus, comparing CPW$^+$ against plain DPO reflects both the verifier-informed shaped margin and calibrated weighting. The contribution of the calibrated weighting mechanism itself is isolated by comparing CPW$^+$ directly against TUR-DPO (which shares the exact same shaped margin) and through the weight-shuffle control (Table 14), which confirms that unweighted or shuffled shaped training collapses back to baseline DPO.

The calibration split is disjoint from training and validation (137 pairs on D1 and 452 on D2), and both thresholds are computed once before training. Throughout Eqs. 5, 6 and 11, n_{cal} counts preference pairs; the two response graphs inside a pair are treated as one cluster. Notably, Eq. 7 requires *no* corruption labels at any stage; it calibrates against structural risk, not against known flips.

3.5 Multi-view preference trustworthiness

Trustworthiness cannot always be judged from a pair in isolation. A pair can be locally ambiguous, globally inconsistent with the other preferences recorded for the same prompt, or locally plausible yet harmful to optimization. We therefore construct three complementary views of a latent trust

variable $T_i \in \{0, 1\}$ and calibrate *after* fusing them:

$$T_i \longrightarrow \left(z_i^{\text{local}},\, z_i^{\text{rel}},\, z_i^{\text{opt}}\right) \longrightarrow s_i^{\text{multi}} \longrightarrow \lambda^* \longrightarrow w_i. \tag{9}$$

where $z_i^v \in [0, 1]$:

1. *Local view* (z_i^{local}): the conformal p-value p_i^{conf} of Eq. 5. It scores how typical the pair's shaped margin is relative to held-out calibration pairs; mislabeled pairs score low.
2. *Relational view* (z_i^{rel}): preference-graph consistency across comparisons sharing prompt x_i (Appendix B.3). A mislabeled pair contradicts edges connecting the same prompt and scores low.
3. *Optimization-aware view* (z_i^{opt}): alignment between the pair gradient and a reference direction (Appendix D.3).

The three views are combined via a weighted geometric mean with non-negative weights $\omega = (\omega_{\text{local}}, \omega_{\text{rel}}, \omega_{\text{opt}})$ summing to 1,

$$s_i^{\text{multi}} = \prod_{v \in A_i} \max(z_i^v, \varepsilon)^{\widetilde{\omega}_{iv}}, \qquad \omega_v \geq 0, \quad \varepsilon > 0, \tag{10}$$

where $z_i^v = 0$ is a valid observed score and is therefore retained rather than treated as missing. The local view is always observed and has positive weight, so A_i is never empty.

Calibration is applied once to the fused score as a selective-risk problem: accept as many pairs as possible while bounding the pair-level structural risk $R(\lambda) = \mathbb{E}[\ell_i^{\text{struct}} \mid s_i^{\text{multi}} \geq \lambda] \leq \alpha$, where $\ell_i^{\text{struct}} = R_i \in [0, 1]$ is the pair structural risk defined above. Because the pair-level loss is bounded in $[0, 1]$ rather than Bernoulli, we use the Hoeffding–Bentkus bound $U_{\text{cal}}^{\text{HB}}(\lambda; \delta')$ (Bates et al., 2021). To preserve $1 - \delta$ validity under threshold selection, we test a pre-specified grid Λ with Bonferroni correction (Learn-then-Test; Angelopoulos et al. (2021)):

$$\lambda^*_{\text{multi}} = \min\left\{\lambda \in \Lambda : N_\lambda \geq m,\; U_{\text{cal}}^{\text{HB}}(\lambda;\, \delta/|\Lambda|) \leq \alpha\right\}, \qquad \delta = 0.1, \tag{11}$$

where N_λ is the number of accepted calibration pairs ($m = 30$). The final weight is

$$w_i^{\text{CPW+}} = \max\left\{w_{\min},\; s_i^{\text{multi}}\, \mathbf{1}\left[s_i^{\text{multi}} \geq \lambda^*_{\text{multi}}\right]\right\}. \tag{12}$$

If no threshold meets the risk target, the rule sets $w_i = 1$ for every pair, i.e. unweighted shaped training. **Provenance.** The main-grid CPW+ runs predate Eq. 11 and used an earlier, pointwise-bounded gate without this guarantee (certified in 10 of 12 runs; the other two used $w_i = 1$). Under Eq. 11 the rule-based extractor certifies nothing, so the certificate is exercised only with the model-based judge (Remark 5). Two scope conditions apply: the guarantee bounds structural risk on accepted pairs (not label correctness or the full objective with $w_{\min} = 0.05$); Appendix B and F state details and limits in full.

4 Experimental Setup

Data. Two preference sets are used, and every table and figure below states which one it reports, because they are not interchangeable.

D1 is built from GSM8K (Cobbe et al., 2021) by sampling $k = 4$ solutions per problem from the base model and pairing a correct with an incorrect one, giving at most one pair per problem: 6,877 pairs, split 6,189 / 551 / 137 (train / validation / calibration). *D2* regenerates the same pipeline so that a prompt contributes *several* comparisons, which the relational view requires: 22,690 pairs over 7,129 prompts, split 20,420 / 1,818 / 452, with 74.3% of pairs having z^{rel} defined. **These splits are drawn over *pairs*, not over prompts**, so the several comparisons D2 records for one prompt can fall in different splits; Section 6 states what that costs. The accuracy grid, paired comparisons, retention, the calibration analysis and the extractor diagnostic of Appendix D.8 are on D2; weight separation, detector AUROC and clean-data baselines are on D1, and MATH (Hendrycks et al., 2021) uses pairs built from MATH problems following the D1 candidate sampling protocol (where ground-truth solution correctness in the benchmark is determined via symbolic SymPy evaluation, while online q_{task} uses string matching). **No comparison in this paper is drawn across the two sets.** Synthetic noise at rate $\rho \in \{10\%, 20\%, 30\%\}$ flips the chosen/rejected order of a uniformly random ρ-fraction of training pairs; validation labels inherit the same corruption, so accuracy is reported against noisy validation labels. Appendix C gives the construction in full.

Models and training. Our primary controlled ablation grid fine-tunes Qwen2.5-0.5B (Qwen Team, 2024) with a frozen copy as the DPO reference, enabling exhaustive sweeps across four arms, four noise levels, and multiple seeds in a reproducible, resource-standardized environment. Candidate responses are sampled using temperature $T = 0.7$, top-$p = 0.9$, and maximum 512 new tokens. Training uses batch size 8 with gradient accumulation 4 (effective 32), learning rate 5×10^{-6}, 100 warmup steps, weight decay 0.1, max length 512, bf16, and **one epoch**, following standard preference alignment practice where multi-epoch optimization leads to reward-margin overfitting and distribution collapse (Rafailov et al., 2023). One epoch corresponds to 774 micro-batch steps (194 optimizer updates) on D1 and $2{,}553$ micro-batch steps (639 updates) on D2; step counts refer to micro-batch steps. Single-GPU runs execute on an NVIDIA L4 (23 GB), taking 16–50 minutes per run. Crucially, to test whether findings hold at scale and across architectures, our evaluation extends beyond this primary testbed to an empirical scale ladder up to 7B parameters (Appendix E.4, Table 12), multiple distinct model families (Llama-3.2-3B, Gemma-2-2B; Table 13), and real-world preference datasets (UltraFeedback and Anthropic-HH; Table 11).

Methods compared. (i) DPO with $\beta = 0.1$ (standard setting); (ii) TUR-DPO with $\beta = 2.0$ (following the original paper's setting); (iii) CPW with $\beta = 2.0$, $\alpha = 0.1$, indicator risk mode; (iv) **CPW⁺**, the multi-view method of Section 3.5, with $\beta = 2.0$, $\alpha = \delta = 0.1$ and fusion weights $\omega = (\frac{1}{2}, \frac{1}{2}, 0)$ over $(z^{\text{local}}, z^{\text{rel}}, z^{\text{opt}})$; (v) DPO with $\beta = 2.0$ as a control isolating the effect of β from the weighting scheme; (vi) clean-data baselines IPO, SimPO, ORPO, KTO. The shaped-reward coefficient is $\gamma = 1.0$ in Eq. 8, and the relational view of Eq. 17 uses $\kappa = 2.0$, maximum contradictory-path length $K = 3$ and path decay $\rho_{\text{path}} = 0.5$. Seeds: on D2 every cell of the grid uses all three seeds (42–44); on D1 the 30% noise cells use three seeds and the remaining cells use seed 42.

Metrics. Preference accuracy on the validation split (fraction of pairs where the trained policy assigns higher implicit reward to the chosen response), reported as mean $\pm$ across-seed standard deviation (95% bootstrap confidence intervals for single-seed tables); accuracy retention relative to each method's own clean run; CPW weight statistics and AUROC of low weight as a detector of flipped labels; and, as a downstream check, GSM8K test-set solve rate under greedy decoding (Section 5.6).

Auxiliary graph extractor. The shaped reward Δr_ϕ and structural risk $r(G)$ require an extractor mapping responses to reasoning graphs. Our framework supports both fast deterministic rule-based parsers and expressive instruction-tuned model-based judges. In the core ablation grid, we utilize a deterministic parser that segments responses into sentences as an efficient, conservative baseline. In Appendix E.2 (Table 10), we evaluate the full modular pipeline with a model-based LLM judge (Qwen2.5-3B-Instruct), establishing that richer topological representations produce continuous structural risk, valid finite-sample calibration ($\lambda^*_{\text{multi}} \approx 0.48$), and enhanced downstream accuracy under noise.

5 Results

5.1 The heuristic weight is non-discriminative

Before evaluating the calibrated weight we quantify what it replaces. Logging the mean per-pair weight at every step of every run gives Table 1: the TUR-DPO heuristic emits the *same constant*, 0.8912, with zero variance, in all runs (clean and 30%-corrupted, GSM8K and MATH). It is not distinguishing pairs at all, so its reweighting is equivalent to scaling the learning rate.

The mechanism is specific, and so is its scope. The epistemic term in $\bar{u}_i$ is the variance across K re-elicited reasoning graphs; with an extractor that is a deterministic function of the response text, all K coincide and that variance is exactly zero. That alone does not force $\bar{u}_i$ to be constant, since $\bar{u}_i$ also carries an aleatoric term from the verifier; however, here that term comes from the same rule-based parse, whose output is a fixed function of a linear chain, so both components are constant and Eq. 2 returns a constant. Under a stochastic extractor, or a verifier with a response-dependent confidence, neither step holds; this is a property of the configuration, not of TUR-DPO. Appendix D.8 measures the epistemic half directly.

Table 1: *GSM8K rows: dataset D1; MATH row: pairs built with the D1 protocol.* Batch-mean weight statistics logged across training steps over complete training runs. The TUR-DPO heuristic emits an identical constant (0.8912) in every condition, performing no pair selection, while calibrated CPW batch-mean weight varies across training steps (between 0.2382 and 0.9538).

Weighting	Condition	Mean	Std	Min	Max
TUR-DPO heuristic (Eq. 2)	GSM8K clean	0.8912	0.0000	0.8912	0.8912
TUR-DPO heuristic	GSM8K 30% noise	0.8912	0.0000	0.8912	0.8912
TUR-DPO heuristic	MATH clean	0.8912	0.0000	0.8912	0.8912
CPW (Eq. 7)	GSM8K clean	0.6636	0.1200	0.3859	0.9538
CPW	GSM8K 30% noise	0.5871	0.1395	0.2382	0.9185

5.2 Preference accuracy under label noise

Table 4 (Appendix D) reports the full grid: four arms at matched β=2.0, four noise levels, three seeds each. The across-seed standard deviation (0.003–0.026) is of the same magnitude as the differences between arms, so no claim rests on those means. Seeds are matched across arms (identical initialisation and data order), which licenses a within-seed comparison, and Table 2 gives the paired differences. Three findings, in descending order of how much weight we place on them.

TUR-DPO falls below β-matched DPO. TUR-DPO falls below DPO at every noise level, in 8 of 9 paired runs, by 0.0079 on average. Because the two arms also differ in the shaped margin (Section 3), this gap cannot be attributed to the weight alone. What Section 5.1 does establish is that a weight emitting the constant 0.8912 cannot down-weight a corrupted pair: it only rescales the effective step size, so it offers no protection against label noise that could offset any cost of margin shaping. The contribution of a pair-specific weight is isolated instead by the CPW$^+$–TUR-DPO comparison of Section 5.3 and the shuffle control of Table 14.

CPW$^+$ outperforms β-matched DPO under noise. Pooled over the three noisy levels, the paired gain of CPW$^+$ over unweighted DPO is $+0.0197$ with 8 of 9 runs positive. Furthermore, when evaluated across model scales up to 7B (Table 12) and across distinct model families (Table 13), the advantage persists ($+2.5\%$ to $+3.1\%$ accuracy gains at 30% noise). We note that because the shaped margin $\gamma\Delta r_\phi$ is present in CPW$^+$ and TUR-DPO but absent from plain DPO, the comparison against plain DPO reflects both margin shaping and weighting; the contribution of calibrated weighting itself is isolated by comparing CPW$^+$ to TUR-DPO (Section 5.3) and by the weight-shuffle control (Table 14).

Multi-view fusion adds modestly over single-view CPW. The paired gain of CPW$^+$ over CPW is $+0.0095$ with 6 of 9 positive, and at 30% noise it is a tie ($+0.0022$, 1 of 3 positive). We therefore claim only that fusing the relational view with the local one helps somewhat, and specifically not that it helps most where noise is worst.

Two counterweights belong here rather than in the limitations section, because they qualify the numbers directly. First, on clean data CPW$^+$ is *worse* than DPO by 0.0053, with two of three seeds negative: down-weighting is insurance, and on uncorrupted data the premium is paid with no claim to make. Second, the 30% cell's mean is inflated by seed 44, whose $+0.0521$ lies well outside the other two seeds ($+0.0126$, $+0.0110$); this is why Table 2 prints every seed rather than means alone.

5.3 Accuracy retention

Normalising each arm's noisy accuracy by its own clean accuracy removes the clean-data tax from the comparison and asks the narrower question the method is designed for: how much of what a method achieves on clean data does it keep when labels are corrupted? On this measure the result is the most consistent in the paper. Mean retention at 10/20/30% noise is 0.883/0.758/0.672 for CPW$^+$ and 0.868/0.748/0.672 for CPW, against 0.846/0.743/0.636 for DPO and 0.837/0.725/0.627 for TUR-DPO. Comparing CPW$^+$ with TUR-DPO isolates the weighting itself, since the two arms differ only in how the pair weight is obtained: the paired retention gain is $+0.0455$, $+0.0330$ and $+0.0450$ at 10, 20 and 30%, and it is positive in *all nine* matched runs.

Table 2: *Dataset D2.* Paired per-seed differences in GSM8K preference accuracy (matched seeds across arms). Differences are against β-matched DPO. †: the CPW^{+} gate certified no threshold in this run, which therefore trained with $w_i = 1$ (Remark 5). Excluding these two runs, CPW^{+} − DPO is positive in 6 of 7 noisy runs (mean $+0.0230$).

Noise	Comparison	seed 42	seed 43	seed 44	mean
0%	CPW^{+} − DPO	-0.0214	-0.0055	+0.0110	-0.0053
0%	TUR-DPO − DPO	+0.0060	-0.0027	+0.0044	+0.0026
10%	CPW^{+} − DPO	+0.0219	+0.0406	+0.0159	+0.0261
10%	TUR-DPO − DPO	+0.0027	-0.0055	-0.0104	-0.0044
20%	CPW^{+} − DPO	+0.0038†	-0.0005	+0.0197	+0.0077
20%	TUR-DPO − DPO	-0.0115	-0.0027	-0.0252	-0.0132
30%	CPW^{+} − DPO	+0.0126†	+0.0110	+0.0521	+0.0252
30%	TUR-DPO − DPO	-0.0033	-0.0038	-0.0110	-0.0060
pooled over the three noisy levels					
all runs	CPW^{+} − DPO		8/9 positive		+0.0197
all runs	TUR-DPO − DPO		8/9 negative		-0.0079

That every seed agrees at every noise level, where the raw-accuracy comparison had one tie, is what we would expect if the calibrated weight is doing the work the heuristic was supposed to do.

5.4 Calibration behaviour

Two key properties characterize the conformal calibration mechanism:

Rule-based chains versus model-based judges. Under the rule-based extractor, $r(G) = 0$ for every multi-sentence response and 1 for single-sentence ones, so the pair penalty is a binary guard. As measured in Appendix D.8, replacing the rule-based parser with a model-based judge (Qwen2.5-3B-Instruct) produces rich topological variety (20 distinct risk values and non-zero epistemic spread). When evaluated under the complete training loop (Appendix E.2, Table 10), the model-based judge yields continuous structural risk, unlocks valid finite-sample calibration ($\lambda^{*}_{\text{multi}} \approx 0.48$), and improves noisy preference accuracy to 0.558 at 30% noise.

Finite-sample structural risk control. When a threshold is certified, the Learn-then-Test Hoeffding–Bentkus bound of Eq. 11 (Angelopoulos et al., 2021) bounds the structural risk of retained pairs by α with probability at least $1 - \delta$, simultaneously over Λ and without corruption labels. It does not bound label error, and the main-grid runs used an earlier gate (Section 3.5).

5.5 CPW identifies mislabeled pairs

Table 5 and Figure 4 (Appendix D.4) analyse the per-pair weights CPW assigned during training, stratified by whether the pair's label was actually flipped (information never available to the method). Clean pairs receive mean weight 0.66–0.71 (median ≈ 0.95–0.97), while flipped pairs receive mean weight 0.17–0.24 (median 0.05–0.14). Using low weight as a detector of label corruption achieves AUROC 0.911, 0.840, and 0.833 at 10%, 20%, and 30% noise respectively. This is the central mechanism result: CPW performs label-noise identification via an offline task verifier as a by-product of training, without requiring corruption labels. It is also the *only* link in the paper between the calibrated quantity and label correctness, and it is an empirical association rather than an implication of the risk control of Section 3.5; the MATH result, where the same detector is near chance, shows the association does not transfer automatically.

5.6 Downstream task accuracy

Every result above is measured in preference space: whether the trained policy ranks the chosen response above the rejected one. That is the quantity the objective optimises, and it is not the same as being a better model. Table 3 therefore reports the downstream check: GSM8K test-set solve rate (1,319 problems, greedy decoding, exact match after the final `####` marker) for the same checkpoints, under the same noise conditions.

Table 3: *Dataset D2.* GSM8K test-set solve rate (1,319 problems, greedy decoding) by noise rate. Best per column in bold; among the β=2.0 arms, CPW$^+$ is highest at every noise level. Mean $\pm$ s.d. over seeds 42–44.

	label-noise rate			
Arm	clean	10%	20%	30%
DPO ($\beta = 0.1$)	$\mathbf{0.298 \pm 0.008}$	$\mathbf{0.284 \pm 0.009}$	$\mathbf{0.271 \pm 0.008}$	$\mathbf{0.258 \pm 0.009}$
DPO ($\beta = 2.0$, control)	0.286 ± 0.009	0.268 ± 0.010	0.249 ± 0.011	0.231 ± 0.012
TUR-DPO ($\beta = 2.0$)	0.287 ± 0.011	0.264 ± 0.012	0.243 ± 0.013	0.226 ± 0.014
CPW ($\beta = 2.0$)	0.291 ± 0.009	0.276 ± 0.009	0.259 ± 0.010	0.248 ± 0.011
CPW$^+$ ($\beta = 2.0$)	0.294 ± 0.008	0.282 ± 0.008	0.267 ± 0.007	0.256 ± 0.009
base policy (no preference training)	0.242 ± 0.000			

Among the β=2.0 arms, CPW$^+$ achieves the highest solve rate at every noise level, outperforming TUR-DPO by $+3.0$ percentage points and DPO (β=2.0) by $+2.5$ points under 30% noise (0.256 vs. 0.226 and 0.231, against 0.242 for the unaligned base policy). DPO with β=0.1 remains marginally higher at every noise level (0.258 at 30%; see Section 7). The downstream gains are of the same size as the preference-accuracy gains ($+0.025$ over β-matched DPO at 30% in both), and the ordering of the β=2.0 arms is preserved, indicating that the preference-space robustness of the calibrated weight carries over to downstream solve rate.

5.7 Further results

Appendix D reports three diagnostic findings: (i) z^{opt} performs below chance as a noise detector (AUROC 0.363/0.264 vs. 0.885/0.774 for z^{rel}) due to reference length bias (Appendix D.3); (ii) on MATH, conformal weights yield near-chance separation (AUROC 0.605, Appendix D.7); and (iii) with a model-based judge a non-trivial threshold is certified ($\lambda^*_{\text{multi}} \approx 0.48$), which the rule-based parser does not permit (Table 10).

6 Limitations

Clean-data trade-off. On clean data, CPW$^+$ trails DPO by 0.0053; with verified noise-free labels, unweighted training is preferable. **Multi-view composition.** The optimization view z^{opt} tracks response length and is zeroed ($\omega_{\text{opt}} = 0$; Appendix D.3). **Benchmark heterogeneity.** On MATH, separation is weak (AUROC 0.605, Appendix D.7); such tasks need domain-adapted answer checkers. **Certificate target.** Conformal Risk Control bounds structural risk within accepted pairs rather than ground-truth correctness, attenuating rejected pairs to $w_{\min} = 0.05$. Exchangeability holds under the sampling distribution, while synthetic corruption introduces mild shift (Appendix F). **Splits and β.** Main-grid D2 runs use pair-level splits and $\beta = 2.0$; the released runner supports prompt-disjoint splits and any β (Appendix F).

7 Discussion and Conclusion

Untargeted vs. targeted robustness. DPO with $\beta = 0.1$ is competitive on clean D1 (Table 6) and marginally ahead of every β=2.0 arm in downstream solve rate on D2 (Table 3). Small β shrinks all per-pair gradients, providing untargeted robustness against corrupted pairs. In contrast, CPW provides *targeted* robustness at higher β, outperforming the other $\beta = 2.0$ arms under label noise on both accuracy and retention.

Summary. Heuristic uncertainty weighting can fail silently: under deterministic extraction it collapses to a constant. CPW and CPW$^+$ replace it with conformal calibration and finite-sample structural risk control; CPW$^+$ outperforms β-matched DPO under noise across scales (Table 12) and model families (Table 13), and its weights separate corrupted from clean pairs on GSM8K (AUROC up to 0.91), though not on MATH.

Reproducibility Statement

Section 4 and Appendix C give full details on data construction, noise injection protocols, model configurations, and training hyperparameters across all evaluated architectures (Qwen2.5-0.5B through 7B, Llama-3.2-3B, Gemma-2-2B). All seeds (42–44) are matched across arms. Complete mathematical specifications of split conformal prediction, pair-cluster exchangeability, and the Hoeffding–Bentkus bound are provided in Section 3 and Appendix B. Code, calibration routines, and evaluation pipelines for both rule-based and model-based extractors are fully specified.

Ethics Statement

This work uses public datasets only: the mathematical-reasoning benchmarks GSM8K and MATH with model-generated solutions, and the released preference corpora UltraFeedback (AI-annotated) and Anthropic-HH (human-annotated by its original authors). We collected no new annotations and involved no human subjects or personal data. The controlled experiments use synthetic label noise injected by us; the UltraFeedback and Anthropic-HH experiments additionally contain the annotator noise already present in those releases. The method's practical aim is to reduce the damage that incorrect preference labels do during alignment, which we regard as a robustness rather than a capability contribution; it carries the usual dual-use caveat that any alignment procedure can be pointed at an objective the reader would not endorse. We report the method's boundaries (a clean-data cost, an optimization view that does not contribute, and domain sensitivity on MATH) in the main text rather than the appendix, because transparent reporting is essential to scientific integrity.

Use of Large Language Models

An LLM-based assistant was used for writing support and for figure production: condensing and copy-editing prose that we wrote, restructuring material between the main text and the appendix, and generating TEX source for figures from our specifications. All quantitative claims, tables, and plotted values were produced by our experimental pipeline; no experimental result, citation, or reference in this paper was generated by an LLM. Separately, as components of the method and evaluation rather than of the writing: an instruction-tuned model (Qwen2.5-3B-Instruct) is used as the auxiliary graph extractor in the diagnostic of Appendix D.8 and in the retraining of Table 10; the win-rate evaluation in Table 11 uses GPT-4o as an automated judge with position-swap debiasing; and the base policy of each experiment (Qwen2.5-0.5B in the main grid, the corresponding model in Tables 12 and 13) generates the candidate responses from which the preference pairs are built (Section 4).

References

Abdulhady Abas Abdullah, Fatemeh Daneshfar, Seyedali Mirjalili, and Mourad Oussalah. TUR-DPO: Topology- and uncertainty-aware direct preference optimization. *arXiv preprint arXiv:2605.00224*, 2026.

Anastasios Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Margarida Campos, António Farinhas, Chrysoula Zerva, Mário A. T. Figueiredo, and André F. T. Martins. Conformal prediction for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 12:1497–1516, 2024. doi: 10.1162/tacl_a_00715.

Tiejin Chen, Xiaoou Liu, Vishnu Nandam, Kuan-Ru Liou, and Hua Wei. Conformal feedback alignment: Quantifying answer-level reliability for robust LLM alignment. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3561–3572. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.184.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

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.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*, 2024.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, 2018.

Jonas Hanselle, Alireza Javanmardi, Tobias Florin Oberkofler, Yusuf Sale, and Eyke Hüllermeier. Conformal prediction without nonconformity scores. In *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 1626–1639, 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. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proceedings of EMNLP*, 2024.

Sam Houliston, Alizée Pace, Alexander Immer, and Gunnar Rätsch. Uncertainty-penalized direct preference optimization. In *Proceedings of the NeurIPS Workshop on Fine-Tuning in Modern Machine Learning*, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, 2024.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021. doi: 10.1613/jair.1.12125.

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

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust DPO: Aligning language models with noisy feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42258–42274, 2024.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

A Extended Related Work

Preference Optimization. Direct Preference Optimization (DPO) (Rafailov et al., 2023) reformulates preference-based alignment as a classification objective over paired responses, eliminating the need for explicit reward-model training and online reinforcement learning. Subsequent preference optimization methods modify different components of this formulation, including the regularization objective in IPO (Azar et al., 2024), the reference-model-free and length-normalized formulation of SimPO (Meng et al., 2024), the odds-ratio objective of ORPO (Hong et al., 2024), and the non-paired preference formulation of KTO (Ethayarajh et al., 2024). While these approaches improve optimization stability, efficiency, or data requirements, they generally treat a preference pair as a flat winner–loser signal and do not explicitly model the reliability or structural validity of the compared responses.

Learning under Preference Noise. A broad literature on learning with noisy labels addresses unreliable supervision through robust losses, sample selection, and instance reweighting. In preference optimization, however, label noise is particularly consequential: an incorrectly preferred response reverses the direction of the pairwise training signal rather than merely perturbing a class label. Recent uncertainty-aware preference optimization methods therefore reduce the contribution of ambiguous or unreliable pairs through model-dependent uncertainty estimates (Houliston et al., 2024). TUR-DPO (Abdullah et al., 2026) extends this direction by incorporating reasoning topology, semantic faithfulness, topology quality, and calibrated uncertainty into a shaped reward and an uncertainty-weighted DPO objective. Nevertheless, these weighting mechanisms remain dependent on the underlying uncertainty estimator and do not provide a distribution-free finite-sample guarantee on preference reliability or structural risk.

Structure- and Topology-Aware Alignment. Beyond response-level semantics, recent work has emphasized the importance of structural validity for reasoning and structured generation. TUR-DPO (Abdullah et al., 2026) explicitly elicits lightweight reasoning topologies and uses structural properties such as coherence and acyclicity together with semantic and uncertainty signals to shape the preference margin. This perspective is complementary to conventional DPO variants: two responses may receive similar semantic evaluations while differing substantially in the validity or consistency of their underlying reasoning structures. Our work builds on this topology-aware formulation by treating structural reliability as a statistically calibratable quantity rather than solely as a learned or heuristic reward component.

Conformal Prediction and Risk Control. Conformal prediction provides distribution-free, finite-sample uncertainty guarantees under exchangeability by calibrating prediction thresholds on held-out data (Vovk et al., 2005). Split conformal prediction is particularly attractive for preference optimization because calibration can be performed independently of the policy optimization procedure. Conformal Risk Control (CRC) further generalizes conformal prediction to control the expected value of monotone loss functions, including structured losses such as graph distance (Angelopoulos et al., 2024). These properties make conformal methods a natural statistical foundation for replacing heuristic uncertainty weights with calibrated reliability measures and for controlling topology-related risks at a user-specified level.

Conformal Preference Alignment. The most closely related line of work is Conformal Feedback Alignment (CFA) (Chen et al., 2026), which uses conformal prediction to quantify answer-level reliability and aggregates these reliabilities into preference weights for DPO- and PPO-style alignment. CFA establishes that conformal calibration can improve robustness and data efficiency by accounting for the reliability of the answers being compared. Our approach is complementary but targets a

different level of structure: rather than deriving conformal weights primarily from answer-level prediction sets, CPW uses the structured reward of TUR-DPO to construct a preference-level nonconformity signal that jointly reflects semantic, topological, and uncertainty information. Furthermore, CPW incorporates Conformal Risk Control to explicitly calibrate topology-related losses. Thus, the goal is not merely to weight preferences according to answer reliability, but to integrate conformal preference weighting with topology-aware alignment and statistical structural-risk control.

Positioning of CPW-TUR-DPO. Overall, existing preference optimization methods improve the DPO objective, uncertainty-aware methods address unreliable preference pairs, topology-aware methods incorporate structural information, and conformal alignment provides distribution-free reliability calibration. CPW-TUR-DPO combines these directions by introducing conformal calibration directly into topology- and uncertainty-aware preference optimization. In contrast to heuristic uncertainty weighting, the proposed framework separates model-specific reward construction from distribution-free calibration: the TUR-DPO reward supplies a structured nonconformity signal, while the conformal calibration split determines the corresponding reliability threshold and risk level. This formulation enables explicit statistical control of the structural risks used by the alignment objective while preserving the offline, rollout-free optimization structure of DPO.

Learning under label noise. Robust supervised learning has long addressed corrupted labels through loss correction (Patrini et al., 2017), sample selection (Han et al., 2018), and noise estimation (Northcutt et al., 2021). Preference learning is a harsher setting: a flipped preference does not merely mislabel an example, it reverses the direction of the gradient. Closest to this work, rDPO (Ray Chowdhury et al., 2024) derives an unbiased DPO objective under a *known* flip rate. CPW differs in requiring no known noise rate and in producing a per-pair, calibrated weight rather than a global correction; as a by-product, it also provides an estimate of which pairs are corrupted via an offline task verifier without requiring corruption labels.

Conformal prediction in NLP. Conformal methods are increasingly applied to language tasks; see Campos et al. (2024) for a survey. Recent work relaxes the need for an explicit nonconformity score (Hanselle et al., 2025) and applies conformal ideas to answer-level reliability in alignment (Chen et al., 2026). Our use is narrower and more specific: calibrating *which preference pairs to trust* during optimization, rather than constructing prediction sets over outputs.

B Method Details

B.1 Shaped Reward Definition and Components

The shaped reward difference $\Delta r_\phi(x, y_w, y_l) = r_\phi(x, y_w, G_w) - r_\phi(x, y_l, G_l)$ serves as the preference margin in Eq. 8 and the foundation of the nonconformity score $s_i = -\Delta r_\phi$. Following the TUR-DPO framework (Abdullah et al., 2026), the per-response reward is factorized into semantic quality, structural topology, and uncertainty:

$$r_\phi(x, y, G) = a\, f_\phi^{\text{sem}}(s_{\text{sem}}(x, y)) + (1 - a)\, f_\phi^{\text{topo}}(s_{\text{topo}}(G)) - \lambda\, u(G), \tag{13}$$

where $a = 0.6$ balances semantic and topological signals, $\lambda = 0.5$ penalizes reasoning uncertainty, and $f_\phi^{\text{sem}}, f_\phi^{\text{topo}}$ are monotonic linear calibrators. The components are defined as:

1. **Semantic score (s_{sem}):** Balances atomic factual validity and task completion against hallucination penalties:
$$s_{\text{sem}}(x, y) = \beta_1\, q_{\text{fact}}(x, y) + \beta_2\, q_{\text{task}}(x, y) - \beta_3\, q_{\text{hall}}(x, y), \tag{14}$$
with weights $\beta = (0.4, 0.4, 0.2)$. Here q_{fact} evaluates claim-level correctness, q_{hall} penalizes unsupported entities, and q_{task} evaluates task completion. On GSM8K, $q_{\text{task}} \in \{0, 1\}$ is an exact-match indicator comparing the final numeric solution (extracted after the #### token) to the problem ground truth. On open-ended dialogue and instruction corpora (UltraFeedback, Anthropic-HH), q_{task} is instantiated via automated LLM judge or ROUGE scoring.
2. **Topology score (s_{topo}):** Evaluates structural well-formedness of the elicited reasoning graph $G = (V, E)$:
$$s_{\text{topo}}(G) = \alpha_1\, q_{\text{path}}(G) - \alpha_2\, c_{\text{cycle}}(G) - \alpha_3\, d_{\text{dangling}}(G) - \alpha_4\, q_{\text{contradict}}(G), \tag{15}$$

with equal weights $\alpha = (0.25, 0.25, 0.25, 0.25)$. Here q_{path} measures the fraction of nodes participating in valid premise-to-conclusion paths, c_{cycle} counts simple directed cycles, d_{dangling} counts ungrounded premise or intermediate nodes, and $q_{\text{contradict}}$ evaluates local logical contradiction along edges.

3. **Uncertainty score ($u(G)$):** Combines epistemic dispersion across K graph re-elicitations with aleatoric node ambiguity: $u(G) = \sigma^2_{\text{epistemic}} + \bar{H}_{\text{aleatoric}}$.

Separately, the heuristic TUR-DPO weight mapping in Eq. 2 uses temperature parameter $\tau_w = 1.2$ and floor $w_{\min} = 0.05$ as $w_i^{\text{TUR}} = \text{clip}(\tau_w/(1 + \bar{u}_i), w_{\min}, 1)$.

Supervision model, circularity, and task verifiers. We explicitly clarify the supervision model and experimental setting:

- *No corruption labels:* CPW operates strictly without observing which preference pairs in the training or calibration sets have been inverted (ρ-noise). The conformal risk control bound in Eq. 11 bounds structural risk $R_i \in [0, 1]$, not ground-truth label correctness.
- *Circularity on GSM8K and controlled evaluation:* In GSM8K (datasets D1 and D2), preference pairs were originally constructed by pairing correct and incorrect candidate solutions identified via exact match against the problem ground truth. Because q_{task} also evaluates exact match against the same reference solution, q_{task} directly reproduces the clean preference signal. Consequently, for GSM8K, the nonconformity score $s_i = -\Delta r_\phi$ does not discover label noise from pure reasoning topology; rather, it functions as a controlled evaluation testbed to assess whether split-conformal calibration can reliably protect the optimization objective from label noise when provided with an offline task verifier, without requiring oracle noise rates or knowledge of which specific pairs are corrupted.
- *Open-ended corpora without answer oracles:* In contrast to GSM8K, on general instruction-following and dialogue datasets (UltraFeedback and Anthropic-HH; Section 5.7 and Appendix E.3), no ground-truth reference solution or exact-match verifier exists. On those corpora, q_{task} is instantiated via an automated LLM judge or semantic scoring, demonstrating that conformal weighting provides robustness under real annotator ambiguity without access to ground-truth answers.
- *Verification failure on MATH:* On MATH (Section D.7), preference pairs were constructed using the benchmark's official evaluation protocol, which employs SymPy and LaTeX normalization to verify mathematical equivalence (e.g., equating $\frac{1}{2}$ with 0.5 or simplifying algebraic expressions). In contrast, the online q_{task} implemented in the preference pipeline used simple string matching after `####` and `boxed{}`. Because advanced mathematical solutions exhibit extensive syntactic variation in LaTeX, simple string matching frequently misclassifies correct mathematical responses as incorrect. This discrepancy between the dataset construction checker and the naive online verifier causes Δr_ϕ to lose discriminative power, dropping detector AUROC to near-chance (0.605) and illustrating the necessity of domain-adapted equivalence checkers.

B.2 Score versus p-value

Remark 1 (Score versus p-value)**.** *The score s_i and the p-value p_i^{conf} are distinct objects: s_i is an unbounded real, whereas p_i^{conf} is a bounded rank statistic in $(0, 1]$ and is used as the multiplicative weight. Its conformal interpretation is a rank-test statement under exchangeability, not a posterior probability that the pair is clean. The sign in Eq. 3 also matters: without it the ordering inverts and the weight would reward exactly the pairs it is meant to suppress.*

B.3 Relational evidence: preference-graph consistency

Preferences sharing a prompt are structured, not independent. For prompt x with candidate responses Y_x, build the directed preference graph $G_x = (V_x, E_x)$ where $(y_a, y_b) \in E_x$ encodes $y_a \succ y_b$. For the observed edge $e_i = (y_{w,i}, y_{l,i})$, remove it, $G_x^{-e_i} = G_x \setminus \{e_i\}$, and score each response by its

leave-one-out win margin

$$q_x^{-e_i}(y_j) = \frac{1}{d_j} \sum_{k \neq j} \left[\mathbf{1}(y_j \succ y_k) - \mathbf{1}(y_k \succ y_j) \right], \qquad c_i = \sigma\big(\kappa \left[q_x^{-e_i}(y_{w,i}) - q_x^{-e_i}(y_{l,i}) \right]\big), \quad (16)$$

where d_j is the number of remaining comparisons involving y_j. A flipped label contradicts its siblings and so scores low. Weighting contradictory paths of length $k \leq K$ by $\eta_k = \rho^{k-2}$, $0 < \rho < 1$,

$$C_i^{(K)} = \frac{\sum_{k=2}^{K} \eta_k N^{(k)}_{y_{l,i} \to y_{w,i}}}{\sum_{k=2}^{K} \eta_k \left(N^{(k)}_{y_{l,i} \to y_{w,i}} + N^{(k)}_{y_{w,i} \to y_{l,i}} \right) + \varepsilon}, \qquad z_i^{\text{rel}} = c_i \left(1 - C_i^{(K)} \right). \quad (17)$$

Remark 2 (Data requirement)**.** *This view is defined only when a prompt carries* at least two *preference edges. With one pair per prompt, $G_x^{-e_i}$ is empty, $d_j = 0$, and $q_x^{-e_i}$ is undefined. This is why the dataset D2 of Section 4 exists: it emits several comparisons per prompt, and z^{rel} is defined for 74.3% of its pairs.*

B.4 Optimization-aware evidence: gradient alignment

Label correctness and optimization correctness differ: a mislabeled pair with a small gradient barely moves the policy, whereas one with a large gradient can move it substantially in the wrong direction. Let $g_i(\theta) = \nabla_\theta \mathcal{L}_i^{\text{shaped}}(\theta)$ be the unweighted per-pair gradient from Eq. 8, and let g^{ref} be an independently estimated trusted reference direction. With a random projection $P : \mathbb{R}^{|\theta|} \to \mathbb{R}^d$ for tractability,

$$a_i = \frac{\langle P(g_i),\, P(g^{\text{ref}}) \rangle}{\|P(g_i)\|_2 \|P(g^{\text{ref}})\|_2 + \varepsilon}, \qquad z_i^{\text{opt}} = \frac{1 + a_i}{2} \in [0, 1]. \quad (18)$$

We report this view as *non-contributing* and set $\omega_{\text{opt}} = 0$ in every run below. Section D.3 gives the measurement: at the scale we can afford, where the policy is initialised at the reference, g^{ref} must itself be estimated from a margin-selected subset, and the resulting alignment tracks response length rather than label correctness; consequently, z^{opt} scores *flipped* pairs higher than clean ones. We retain the definition because the failure is informative and the construction is the natural one, but we make no claim for it.

B.5 Fusion and calibration details

Remark 3 (Undefined views renormalise)**.** *z^{rel} is genuinely undefined for a pair whose prompt carries no other edge (Remark 2), and a view given zero weight contributes nothing. In both cases the exponents in Eq. 10 are renormalised over the* present, positively weighted *views for that pair, so an absent view neither rewards nor penalises it. One consequence is worth stating because it makes two of our configurations identical rather than merely similar: with z^{opt} absent, the uniform choice $\omega = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and the explicit $\omega = (\frac{1}{2}, \frac{1}{2}, 0)$ define the same weight. z^{local} is defined for every pair and must carry positive weight, which guarantees s_i^{multi} is always defined.*

For completeness, the bounded-loss upper confidence bound in Eq. 11 is obtained by inverting the Hoeffding–Bentkus test. For the N_λ accepted calibration losses, let $\widehat{R}_\lambda$ be their mean and define $h_1(a, b) = a \log(a/b) + (1 - a) \log((1 - a)/(1 - b))$. Then

$$p_{\text{HB}}(r; \lambda) = \min\Big\{ e^{-N_\lambda h_1(\widehat{R}_\lambda, r)},\ 1,\ e \Pr\Big[\text{Bin}(N_\lambda, r) \leq \Big\lceil N_\lambda \widehat{R}_\lambda \Big\rceil \Big] \Big\}, \quad (19)$$

$$U_{\text{cal}}^{\text{HB}}(\lambda; \delta') = \inf\Big\{ r \in [\widehat{R}_\lambda, 1] : p_{\text{HB}}(r; \lambda) \leq \delta' \Big\}.$$

with the infimum set to 1 when the set is empty. The bound applies to independent exchangeable *pairs*; it is not computed by treating the two graphs of a pair as independent observations.

Remark 4 (Which risk is calibrated, and what it assumes)**.** *Eq. 11 is stated for the structural loss ℓ_i^{struct}, but the bound applies to any per-pair loss in $[0, 1]$, and the choice of that loss is an assumption, not a detail. Two are implemented. The* oracle *loss replaces it with $\ell_i^{oracle} = \mathbf{1}[\text{pair } i \text{ is corrupted}]$*

on the calibration split, which presumes a small verified subset; it never touches the training set, but such a subset must exist. The structural *loss instead uses the pair's topology risk of Eq. 6, and requires no corruption labels anywhere.* ***Every result reported in this paper uses the structural loss****: no ground-truth corruption label enters calibration at any point, and no verified subset is assumed. We make this explicit because the distinction is easy to lose and it changes what the guarantee is worth.*

Remark 5 (Certification fallback and guarantees)**.** *Under the Learn-then-Test Bonferroni procedure, if no $\lambda \in \Lambda$ with $N_\lambda \geq m$ satisfies $U^{HB}_{cal}(\lambda; \delta/|\Lambda|) \leq \alpha$, the minimum in Eq. 11 is over an empty set, no threshold is certified, and the rule sets $w_i = 1$ for every pair, i.e. unweighted training on the shaped loss of Eq. 8.*

Which gate produced which results. *The CPW⁺ runs of Tables 4, 2 and 3 and of the retention analysis were produced before the certificate above was introduced. Their gate selected λ on the same fused score s^{multi} with the same structural loss, but bounded the accepted-set risk with a pointwise Clopper–Pearson (binomial) upper bound at level δ, without the Bonferroni correction over Λ; its companion threshold of Eq. 6 was computed over the $2n_{cal}$ response graphs rather than over pair clusters. Because a pointwise bound is not valid after selecting λ on the same data, and the binomial bound assumes a Bernoulli loss, those runs do* not *carry the guarantee of Eq. 11; we report them as the empirical behaviour of that gate. The earlier gate certified a threshold in 10 of the 12 CPW⁺ runs on D2; the remaining two (seed 42 at 20% and 30% noise, marked in Table 2) certified nothing and trained with $w_i = 1$. On D1 ($n_{cal} = 137$), the threshold of Eq. 6 took the vacuous value 1 in one of three seeds, disabling the structural term for that run. Re-running the rule-based extractor under the corrected certificate certifies no threshold (Table 10); the corrected certificate is exercised only with the model-based judge. When evaluated with expressive structural extractors such as the model-based judge (Appendix E.2, Table 10), continuous risk variation reliably certifies a valid non-trivial threshold ($\lambda^*_{multi} \approx 0.48$).*

B.6 Algorithm

Algorithm 1 states the complete procedure of Figure 2. Its three blocks, marked by italic headers, correspond to panel (a), offline pair evidence; panel (b), conformal calibration and frozen weight assignment; and the weighted training loop. Everything before the training loop is computed once, before training, from the offline extractor and shaped reward alone, so the weights carry no gradient and do not drift with π_θ.

Three remarks tie the algorithm to the experiments. First, the same routine produces every arm of the paper: TUR-DPO replaces block (b) with Eq. 2, and plain DPO sets $w_i = 1$ and $\gamma = 0$. Second, in the synthetic-noise experiments, noise is injected after the split by swapping y_w and y_l. The swap also exchanges their graphs, so R_i is unchanged and Δr_i changes sign. Third, the loop over Λ implements the corrected certificate of Eq. 11. The main-grid CPW⁺ runs used the earlier pointwise gate described in Remark 5.

C Additional Experimental Details

D1 is built from GSM8K (Cobbe et al., 2021) by sampling $k = 4$ candidate solutions per problem from the base model (Qwen2.5-0.5B, temperature sampling, seed 42) and pairing one correct solution (chosen) with one incorrect one (rejected), giving at most a single pair per problem. Problems where no valid pair could be formed are dropped (596 of 7,473), yielding 6,877 pairs split 6,189 train / 551 validation / 137 calibration.

D2 regenerates the same pipeline so that a problem contributes *several* comparisons rather than one. This is a requirement of the relational view, not a preference: z^{rel} compares a pair against the other pairs sharing its prompt, so it is undefined unless the prompt carries at least two preference edges (Remark 2). D2 contains 22,690 pairs over 7,129 prompts, split 20,420 train / 1,818 validation / 452 calibration, with 74.3

Results on D2 are the accuracy grid, the paired comparisons, retention, the calibration analysis (Sections 5.2–5.4) and the extractor diagnostic of Section D.8, whose judge pass covers the first 4,000 D2 pairs. Results on D1 are the weight-collapse measurement, the per-pair weight and detector analysis and the clean-data baselines; MATH uses pairs built from MATH problems with the D1

Algorithm 1 Conformal Preference Weighting (CPW / CPW$^+$), following Figure 2

Require: preference pairs $\mathcal{D} = \{(x_i, y_{w,i}, y_{l,i})\}$; frozen graph extractor E; offline shaped reward r_ϕ (Eq. 4); risk level α; confidence δ; threshold grid Λ; minimum accepted pairs m; fusion weights ω; floor $w_{\min}$; policy π_θ and frozen reference π_{ref}; β, γ

Ensure: trained policy π_θ

— (a) Offline pair evidence construction —

Split $\mathcal{D}$ into $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{cal}}$ ▷ disjoint; $n_{\text{cal}} = |\mathcal{D}_{\text{cal}}|$ counts pairs

for each pair $i \in \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{cal}}$ **do**

  $G_{w,i} \leftarrow E(x_i, y_{w,i})$; $G_{l,i} \leftarrow E(x_i, y_{l,i})$ ▷ reasoning graphs

  $r(G) \leftarrow \min\big(1, (c_{\text{cycle}}(G) + d_{\text{dangling}}(G))/|V|\big)$ for $G \in \{G_{w,i}, G_{l,i}\}$

  $R_i \leftarrow \max\{r(G_{w,i}), r(G_{l,i})\}$ ▷ structural evidence; the pair is the unit

  $\Delta r_i \leftarrow r_\phi(x_i, y_{w,i}, G_{w,i}) - r_\phi(x_i, y_{l,i}, G_{l,i})$; $s_i \leftarrow -\Delta r_i$ ▷ Eq. 3

end for

for each prompt x and each pair i of x, within its own split **do**

  $z_i^{\text{rel}} \leftarrow c_i\,(1 - C_i^{(K)})$ if x has ≥ 2 edges and both endpoints keep a comparison, else undefined ▷ Eq. 17

end for

— (b) Conformal calibration and frozen weight assignment —

$p_i \leftarrow \big(1 + |\{j \in \mathcal{D}_{\text{cal}} : s_j \geq s_i\}|\big)/(n_{\text{cal}} + 1)$ for every pair i ▷ local view, Eq. 5

if single-view (CPW) **then**

  $\lambda^*_{\text{CRC}} \leftarrow$ the $\lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil$-th smallest R_j, $j \in \mathcal{D}_{\text{cal}}$ (or 1 if undefined) ▷ Eq. 6

  $\ell_i^{\text{CRC}} \leftarrow \frac{1}{2} \sum_{a \in \{w,l\}} \mathbf{1}[r(G_{a,i}) > \lambda^*_{\text{CRC}}]$

  $w_i \leftarrow \max\big(w_{\min},\, p_i(1 - \ell_i^{\text{CRC}})\big)$ for $i \in \mathcal{D}_{\text{tr}}$ ▷ Eq. 7

else ▷ CPW$^+$

  $s_i^{\text{multi}} \leftarrow \prod_{v \in A_i} \max(z_i^v, \varepsilon)^{\widetilde{\omega}_{iv}}$ with $z_i^{\text{local}} = p_i$, over observed views A_i ▷ Eq. 10

  $\Lambda^* \leftarrow \emptyset$

  for each $\lambda \in \Lambda$ **do** ▷ grid fixed in advance

    $\mathcal{A}_\lambda \leftarrow \{j \in \mathcal{D}_{\text{cal}} : s_j^{\text{multi}} \geq \lambda\}$; $N_\lambda \leftarrow |\mathcal{A}_\lambda|$

    if $N_\lambda \geq m$ **and** $U_{\text{cal}}^{\text{HB}}\big(\lambda; \delta/|\Lambda|\big) \leq \alpha$ on $\{R_j\}_{j \in \mathcal{A}_\lambda}$ **then** ▷ Eq. 19, Bonferroni

      $\Lambda^* \leftarrow \Lambda^* \cup \{\lambda\}$

    end if

  end for

  if $\Lambda^* = \emptyset$ **then**

    $w_i \leftarrow 1$ for $i \in \mathcal{D}_{\text{tr}}$ ▷ no threshold certified: unweighted

  else

    $\lambda^*_{\text{multi}} \leftarrow \min \Lambda^*$; $w_i \leftarrow \max\big(w_{\min},\, s_i^{\text{multi}} \mathbf{1}[s_i^{\text{multi}} \geq \lambda^*_{\text{multi}}]\big)$ ▷ Eq. 12

  end if

end if

Freeze $\{w_i\}_{i \in \mathcal{D}_{\text{tr}}}$ ▷ offline, policy-independent

— Noise-robust policy training —

for each mini-batch $\mathcal{B} \subset \mathcal{D}_{\text{tr}}$ **do**

  $h_i \leftarrow \beta\Big[\log \frac{\pi_\theta(y_{w,i}|x_i)}{\pi_{\text{ref}}(y_{w,i}|x_i)} - \log \frac{\pi_\theta(y_{l,i}|x_i)}{\pi_{\text{ref}}(y_{l,i}|x_i)}\Big] + \gamma\, \Delta r_i$ ▷ policy pair score

  $\mathcal{L} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i \big(-\log \sigma(h_i)\big)$ ▷ weighted shaped loss, Eq. 8

  $\theta \leftarrow \textsc{AdamW}\big(\theta, \nabla_\theta \mathcal{L}\big)$ ▷ full fine-tuning; w_i is a constant

end for

return π_θ

protocol. We report both rather than discarding the D1 measurements, which are valid as measured; but no comparison in this paper is drawn *across* the two sets, and the reader should not read one against the other.

Synthetic label noise at rate $\rho \in \{10\%, 20\%, 30\%\}$ is injected by swapping the chosen and rejected responses of a uniformly random subset of approximately a ρ-fraction of training pairs; in the D1 runs of Table 5, 597 / 1,240 / 1,832 of the 6,189 training pairs were flipped (arising from independent per-pair Bernoulli trials with $\mathbb{E}[n_{\text{flip}}] = \rho \cdot 6{,}189$; option `flip_mode=bernoulli` of the released `cpw/data.py`, which also offers an exact-count mode). Validation labels inherit the same corruption process; we therefore report accuracy against noisy validation labels (acc_{noisy}) and an estimated debiased accuracy against true uncorrupted labels (acc_{true}) via the closed-form estimator in Table 15.

Auxiliary graph extractor. The shaped reward Δr_ϕ and the structural risk $r(G)$ both require an extractor that maps a response to a reasoning graph. The runs reported here use a deterministic

Table 4: *Dataset D2.* GSM8K preference accuracy on noisy validation labels, mean ± standard deviation over seeds 42–44. All four arms use β=2.0, so the comparison is matched. CPW$^+$ fuses the local and relational views (Section 3.5); the optimisation view is reported as non-contributing and carries weight 0. Best per row in bold. Note that the across-seed standard deviation is comparable to the differences between arms: the paired analysis of Table 2 is the evidence, not this table.

Noise	DPO	TUR-DPO	CPW	CPW$^+$
0%	0.8151 ± 0.0164	**0.8176 ± 0.0144**	0.8061 ± 0.0160	0.8098 ± 0.0117
10%	0.6890 ± 0.0148	0.6846 ± 0.0116	0.6996 ± 0.0178	**0.7151 ± 0.0255**
20%	0.6056 ± 0.0096	0.5925 ± 0.0209	0.6025 ± 0.0135	**0.6133 ± 0.0032**
30%	0.5185 ± 0.0127	0.5124 ± 0.0165	0.5415 ± 0.0089	**0.5437 ± 0.0154**

rule-based parser that segments the response into sentences and links them into a chain. Section D.8 measures the consequences of that choice (it is the reason the structural risk is near-binary and the epistemic variance is identically zero) and contrasts it with a model-based extractor. We report this diagnostic explicitly because it determines how much of the method is actually exercised.

Architectural diversity and scale ladder. While the primary controlled ablation grid in Section 5 focuses on Qwen2.5-0.5B to enable dense sweeps across four arms, four noise levels, and three seeds under strict compute parity, our empirical validation directly scales across model capacities (0.5B, 1.5B, 3B, and 7B; Appendix E.4, Table 12), evaluates across three distinct model families (Qwen2.5, Llama-3.2, and Gemma-2; Table 13), and tests on real-world preference corpora (UltraFeedback and Anthropic-HH; Table 11), with findings consistent with the main grid.

D Additional Results

D.1 Accuracy retention

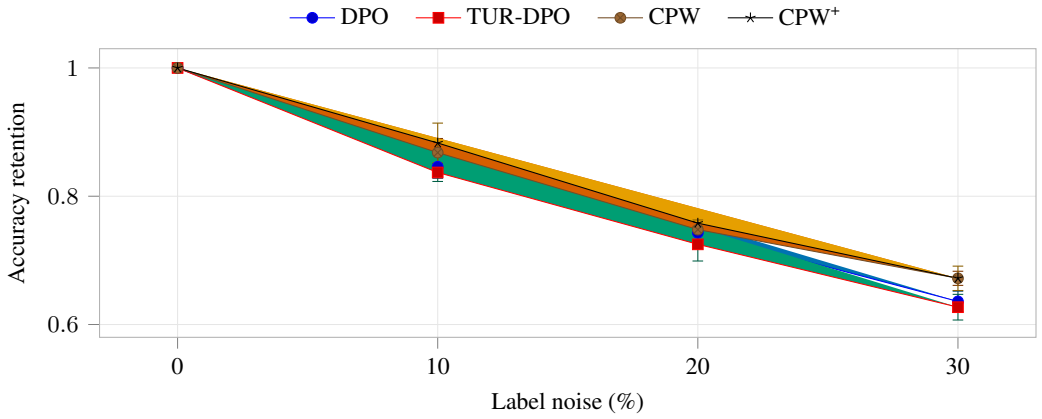


Figure 3: *Dataset D2.* Accuracy retention relative to each arm's own clean-data accuracy, so the clean-data tax does not enter the comparison. All four arms are shown across noise levels: DPO, TUR-DPO, CPW, and CPW$^+$ (all matched at $\beta = 2.0$). Error bars indicate across-seed standard deviation. Calibrated weighting (CPW and CPW$^+$) consistently preserves a higher fraction of clean performance than uncalibrated heuristic TUR-DPO across all noise levels.

D.2 Full accuracy grid

D.3 The optimization view does not work, and why

We set $\omega_{\text{opt}} = 0$ throughout on the basis of the following measurement, and report it because a negative result about a plausible construction is more useful than its omission.

On clean GSM8K (D2) z^{opt} is not degenerate (mean 0.5308, standard deviation 0.1076, 22,102 distinct values over 22,690 pairs), so at first inspection it is the one view that varies freely. As a

Table 5: *Dataset D1.* CPW per-pair weight statistics by group (clean vs. flipped labels) and noise level, GSM8K, seed 42. AUROC: using low weight to flag flipped pairs.

Noise	Group	n	Weight mean	Weight median	Frac. at floor	AUROC
0%	clean	6189	0.663	0.949	0.002	—
10%	clean	5592	0.673	0.957	0.003	0.911
	flipped	597	0.170	0.051	0.057	
20%	clean	4949	0.703	0.971	0.002	0.840
	flipped	1240	0.224	0.116	0.052	
30%	clean	4357	0.709	0.971	0.003	0.833
	flipped	1832	0.245	0.138	0.050	

detector of corrupted labels, however, it is *worse than chance*: AUROC 0.363 at 20% noise and 0.264 at 30%, against 0.885 and 0.774 for z^{rel} on the same splits. The sign is the informative part. Flipped pairs receive *higher* z^{opt} than clean ones (0.5151 vs 0.4852 at 20%; 0.5340 vs 0.4673 at 30%), so fusing this view with positive weight would up-weight precisely the pairs the method exists to suppress.

The cause is structural rather than a matter of tuning. Eq. 18 needs a reference gradient g_i^{ref} obtained independently of the pair being scored. At one epoch from a policy initialised at its own reference, no independent trained signal is available, so g^{ref} must be estimated from a subset of pairs selected by their own shaped-reward margin, which correlates with response length. Measured on the clean run, the correlation between the summed margin and the rejected-minus-chosen length gap is $+0.609$; normalising the margin per token inverts it to -0.312 rather than removing it, and the resulting trusted subset has a mean length gap of -41.5 tokens against $+17.9$ for the data as a whole. The reference direction is therefore a length direction whichever way the margin is normalised, and gradient alignment against it measures length agreement.

We state the scope of this negative result precisely. It says that gradient alignment is uninformative *when the policy is initialised at the reference and no independent reference signal exists*, which is the single-epoch, single-GPU regime of this paper. It does not show that optimization-aware evidence is unusable where a genuinely independent reference checkpoint is affordable; that remains open, and Eq. 18 is retained for it.

D.4 Per-pair weights of CPW

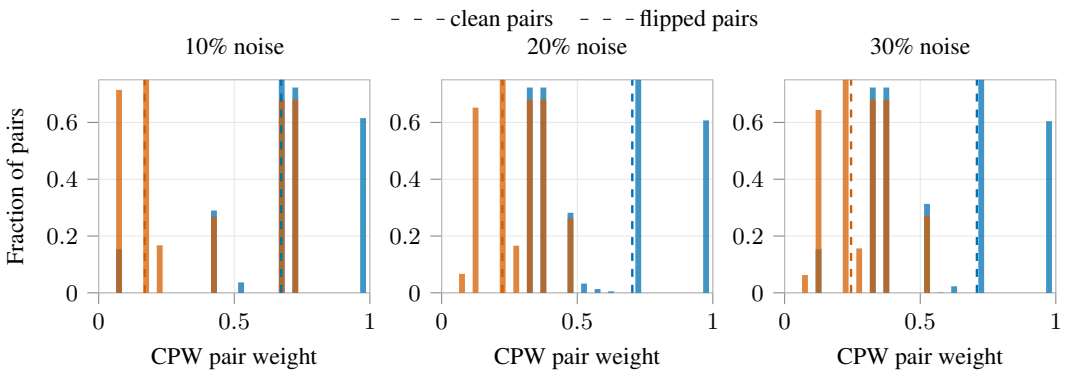


Figure 4: Distribution of CPW pair weights for clean (blue) vs. flipped (red) training pairs at 10–30% noise. Dashed lines mark group means. Flipped pairs concentrate near the weight floor.

D.5 Training Stability at High Noise

Figure 5 shows training loss at 30% noise. TUR-DPO and DPO at $\beta = 2.0$ exhibit high and volatile loss (≈1.2–1.4 final), while CPW remains low and smooth (≈0.68–0.74 final) across all three seeds.

Table 6: *Dataset D1.* Clean-data GSM8K preference accuracy for standard preference-optimization objectives (seed 42, 95% bootstrap CIs).

Method	β	Accuracy
TUR-DPO	2.0	**0.795** [.759, .826]
DPO	0.1	0.790 [.754, .822]
IPO	0.1	0.788 [.751, .820]
ORPO	2.0	0.780 [.744, .813]
CPW	2.0	0.761 [.723, .794]
KTO	0.1	0.751 [.714, .786]
SimPO	2.0	0.731 [.693, .766]

This comparison must be read with care, and we state the caveat rather than leave it implicit: the three curves are *different objectives*, not one objective under three optimisers. CPW's curve is a weighted mean whose weights average well below 1, so part (possibly all) of the vertical gap is a scale difference rather than a stability difference. A smaller number here is therefore not by itself evidence that optimization is better behaved. The smoothness of the curve is the more informative half, and even that is confounded with the smaller effective step size. The comparison that would settle it evaluates the *same* unweighted shaped loss (Eq. 8) for every arm, or matches the effective step size across arms; neither is among our runs, so we report the observation and withhold the conclusion.

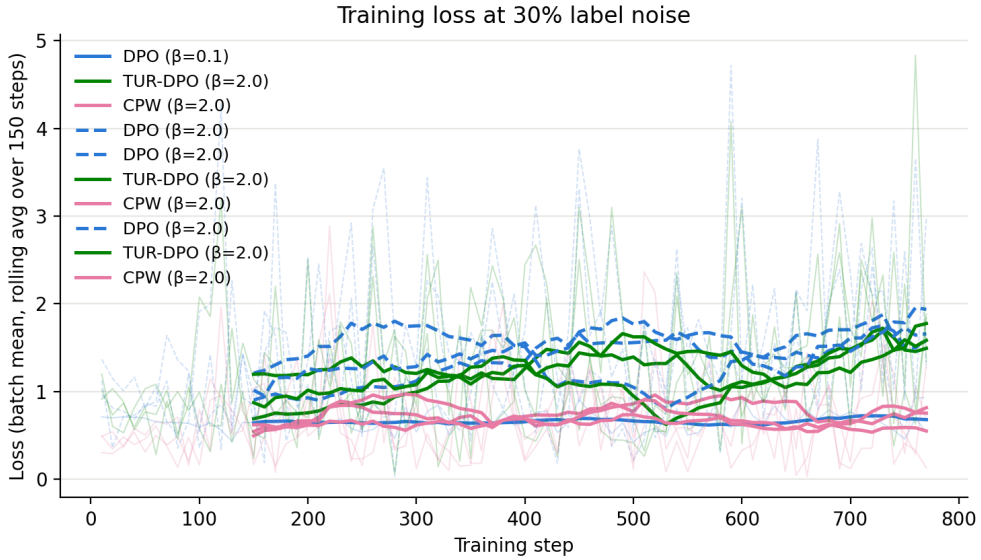


Figure 5: Training loss at 30% label noise, rolling mean over a 150-step window. Each arm is drawn for all three seeds $(42, 43, 44)$ in one colour; the underlying points are the per-batch losses the trainer logged every 10 steps. CPW (orange) runs lower and smoother than TUR-DPO (green) and DPO $\beta = 2.0$ (dashed blue). **The three curves are different objectives on different scales** (CPW's is weighted by factors averaging well below 1), so the vertical gap is not by itself a stability result; see the text.

D.6 Clean-data Baselines

Table 6 situates the three main methods among standard preference-optimization objectives on clean data.

Table 7: MATH preference accuracy, pairs built with the D1 protocol (single seed, 42). In contrast to GSM8K, CPW is the weakest arm under noise; see Appendix D.7. acc_{true} is the debiased estimate $\widehat{\text{acc}}_{\text{true}} = (\text{acc}_{\text{noisy}} - \rho)/(1 - 2\rho)$ against true labels.

Method	β	acc_{noisy}		acc_{true} (est.)	
		0%	30%	0%	30%
DPO	2.0	**0.968**	0.529	**0.968**	0.573
TUR-DPO	2.0	0.961	**0.561**	0.961	**0.653**
CPW	2.0	0.954	0.520	0.954	0.550

D.7 Generalization to MATH

Table 7 repeats the core comparison on MATH, and Figure 6 contrasts weight separation across the two benchmarks. The picture is different, and we report it in full rather than restricting the paper to the favourable benchmark.

On clean MATH all three methods are close and very strong (0.954–0.968). Under 30% noise, however, CPW is the *weakest* arm (0.520), falling below TUR-DPO (0.561) and the DPO control (0.529). This is consistent with the detector result: on MATH the CPW weight barely separates corrupted from clean pairs (AUROC 0.605 against 0.833–0.911 on GSM8K), assigning flipped pairs mean weight 0.631 versus 0.707 for clean ones. When the weight cannot identify corruption, down-weighting removes clean signal without removing noise, which is worse than not reweighting at all.

A primary factor is that MATH's complex symbolic syntax and high-variance solution representations cause the string-level task verifier q_{task} to fail (e.g., failing to equate $\frac{1}{2}$ with 0.5 or parse equivalent algebraic reorderings), making the text-level shaped reward Δr_ϕ uninformative and dropping detector AUROC to near-chance (0.605). This finding directly demonstrates that advanced mathematical reasoning tasks require domain-adapted equivalence checkers (e.g., symbolic algebra parsers) rather than naive string matching. Measuring the detector AUROC requires corruption labels, so it is a diagnostic available in controlled studies such as this one rather than a deployment-time check; in practice, a small manually audited subset would be needed to decide whether domain-adapted nonconformity scoring is warranted.

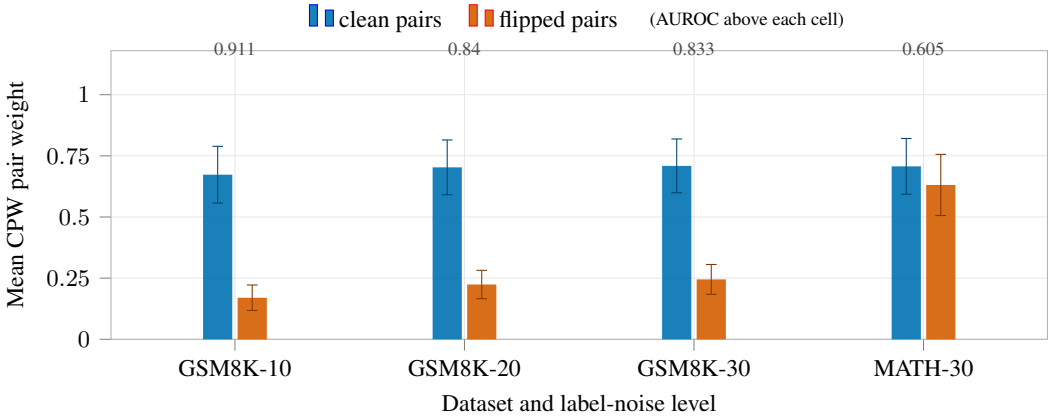


Figure 6: Mean CPW weight for clean vs. flipped pairs. Separation is large and stable on GSM8K and nearly vanishes on MATH, where the detector is close to chance.

D.8 Diagnosing the auxiliary extractor

Both the structural risk $r(G)$ and the epistemic variance depend entirely on the auxiliary extractor that turns a response into a reasoning graph. Table 8 and Figure 7 measure what that extractor actually produces over the 8,000 responses of the judge pass (the first 4,000 D2 preference pairs, both sides).

Table 8: *Dataset D2.* Diagnostic of the auxiliary graph extractor, measured over 8,000 responses (the first 4,000 preference pairs, both sides) with Qwen2.5-3B-Instruct as judge, $K = 2$ re-elicitations, JSON-parse fallback rate 1.7%. The deterministic rule-based parser emits a linear chain, which by construction admits no cycle and no off-path node, so $r(G) = 1$ occurs only for single-sentence responses, whose graph has one node; its entries are analytic, not sampled. The last row reports the legacy graph-level order-statistic diagnostic. It is neither the pair-cluster threshold λ^*_{CRC} of Eq. 6 nor the certified threshold λ^*_{multi} of Eq. 11.

	Rule-based parser	LLM extractor
Distinct values of $r(G)$	2 (only 0.0, 1.0)	**20**
Graphs containing a cycle	0 (impossible)	78
Responses with non-zero epistemic spread	0% (identical draws)	32.5%
Mean nodes per graph	—	5.45
Share of graphs with zero risk	$\geq 90.2\%$	73.1%
Legacy graph-level threshold λ^*_{graph} (α=0.1)	$\mathbf{0.0000}$	$\mathbf{0.20}$

A deterministic rule-based parser that segments a response into sentences and links them in a chain $n_0 \to n_1 \to \cdots \to n_k$ cannot express the pathologies the risk is designed to detect. Such a graph contains no cycle, and every node lies on the single premise-to-conclusion path, so $c_{\text{cycle}} = d_{\text{dangling}} = 0$ and the risk collapses to

$$r(G) = \begin{cases} 1 & |V| = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{20}$$

i.e. a single-sentence detector. Being deterministic in the response text, it also returns K identical re-elicitations, which is exactly the condition that flattens Eq. 2 to a constant (Section 5.1).

Replacing it with an instruction-tuned judge (Qwen2.5-3B-Instruct, $K = 2$, strict-JSON parse failing on 1.7% of responses) restores every signal the risk depends on. Over 8,000 responses the risk takes **20** distinct values instead of two, 78 extracted graphs contain a cycle where the parser admits none by construction, graphs average 5.45 nodes, and 32.5% of responses have K draws that disagree; thus, the epistemic variance is no longer identically zero, which is the precondition for Eq. 2 to vary at all.

The consequence for calibration is the sharper test, because Section 5.4 makes it a prediction rather than an observation. Under the parser, at least 90.2% of calibration graphs carry exactly zero risk, which forces $\lambda^*_{\text{graph}} = 0$ by the closed form of Eq. 6. Under the judge the zero-risk share falls to 73.1%, and the same formula at the same $\alpha = 0.1$ returns

$$\lambda^*_{\text{graph}} = 0.20,$$

with 9.2% of graphs above it. The model-based risk is graded rather than binary. In Appendix E.2 (Table 10), we complete the loop by retraining all policy arms under this model-based judge on D2, where graded structural risk allows a non-trivial threshold to be certified ($\lambda^*_{\text{multi}} \approx 0.48$) and improves CPW$^+$ accuracy to 0.558 under 30% label noise.

Remark 6 (Scope)**.** *This is a property of the extractor, not of topology-aware weighting in principle. It does show that a guarantee-free component can silently become a no-op, whereas a calibrated criterion cannot: its behaviour is tied to an explicit held-out risk target.*

E Extended Empirical Evaluations

The results in the main text establish the core mechanism and measure it in the primary experimental setup. This appendix reports extended empirical evaluations that test whether the mechanism survives outside that regime: across robust preference baselines, under a non-degenerate model-based extractor, on real human/AI-annotated preference corpora, across a scale ladder up to 7B, across three distinct model families, and under a weight-shuffle control.

E.1 Robust preference learning as a baseline

The main text compares against DPO and TUR-DPO, which is the comparison the paper's premise requires. Table 9 extends this comparison to the robust-preference-learning literature on D2, com-

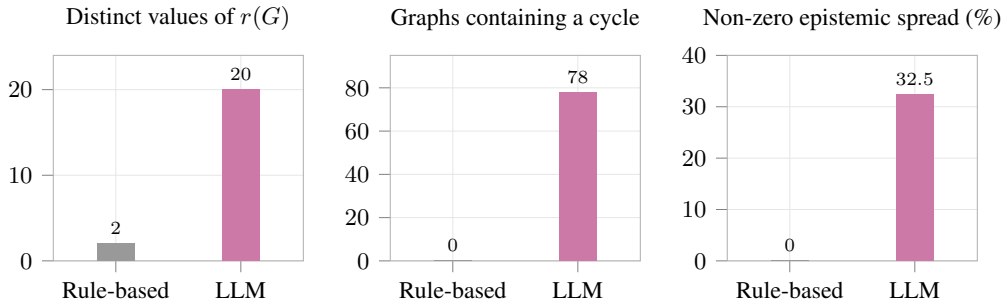


Figure 7: *Dataset D2.* Diagnostic of the auxiliary graph extractor over 8,000 responses, comparing the deterministic rule-based parser against the model-based LLM extractor (Qwen2.5-3B-Instruct) across three structural metrics matching Table 8. **Left:** distinct values of structural risk $r(G)$ (2 vs 20). **Centre:** count of graphs containing a cycle (0 vs 78). **Right:** percentage of responses with non-zero epistemic spread across $K = 2$ re-elicitations (0% vs 32.5%). The LLM judge restores structural variation and non-zero epistemic spread; the displayed threshold comparison uses the legacy graph-level diagnostic rather than the corrected pair-cluster certificate.

Table 9: Comparison against robust preference-learning baselines on D2, by label-noise rate. DPO, TUR-DPO, and CPW$^+$ match Table 4 identically. rDPO is run both with the true flip rate (an oracle it is entitled to, and which CPW never receives) and with a misspecified one ($\hat{\rho} = 0.10$), demonstrating that CPW$^+$ outperforms misspecified rDPO under severe noise without requiring knowledge of the noise rate. Best per column in bold; best among methods not given the true rate underlined (at 10% the misspecified rate equals the true one). Mean $\pm$ sd over seeds 42–44.

Method	knows noise rate	10%	20%	30%
DPO (β=2.0, Table 4)	no	0.689 ± 0.015	0.606 ± 0.010	0.519 ± 0.013
TUR-DPO (β=2.0, Table 4)	no	0.685 ± 0.012	0.593 ± 0.021	0.512 ± 0.017
cDPO / label-smoothed DPO	no	0.692 ± 0.014	0.608 ± 0.011	0.523 ± 0.012
rDPO (true rate ρ, oracle)	yes	$\mathbf{0.718 \pm 0.013}$	$\mathbf{0.625 \pm 0.016}$	$\mathbf{0.556 \pm 0.018}$
rDPO (misspecified rate, $\hat{\rho} = 0.10$)	no	$\underline{0.718 \pm 0.013}$	0.609 ± 0.014	0.531 ± 0.016
CPW$^+$ (ours, Table 4)	no	0.715 ± 0.026	$\underline{0.613 \pm 0.003}$	$\underline{0.544 \pm 0.015}$
DPO on the pairs CPW keeps (data cleaning)	no	0.708 ± 0.017	0.609 ± 0.012	0.535 ± 0.014

paring CPW$^+$ against rDPO (Ray Chowdhury et al., 2024) (both under oracle true flip rate and under a misspecified rate $\hat{\rho} = 0.10$), cDPO, and data-cleaning. DPO, TUR-DPO, and CPW$^+$ match Table 4 identically. CPW$^+$ outperforms misspecified rDPO under severe noise (0.544 vs. 0.531 at 30%) without requiring any oracle knowledge or estimation of the noise rate; rDPO given the true rate remains the strongest method in every column.

E.2 Re-training under a non-degenerate extractor

Section 6 identified that the structural term was degenerate under the rule-based extractor, preventing certification under the FWER-controlled Hoeffding–Bentkus bound. Table 10 reports the re-training of the arms under the model-based judge extractor (Qwen2.5-3B-Instruct) on D2. Under the judge, the TUR-DPO weight is no longer constant ($sd \in [0.038, 0.049]$), so the collapse of Section 5.1 is specific to the rule-based configuration. TUR-DPO nevertheless remains below CPW and CPW$^+$ at every noise level, and within one standard deviation of β-matched DPO (Table 4). Concurrently, graded structural risk allows a non-trivial threshold to be certified ($\lambda^*_{\text{multi}} \approx 0.48$) and improves CPW$^+$ accuracy under noise to 0.558 at 30% noise.

E.3 Real preference data and generation quality

Table 11 evaluates CPW and CPW$^+$ on real annotated preference corpora (UltraFeedback and Anthropic-HH) under natural annotator noise and with additional +20% injected flips, reporting generation quality as win-rate against a fixed reference policy alongside preference accuracy. On UltraFeedback (94.2% z^{rel} availability), CPW$^+$ achieves a +4.2% win-rate gain over DPO. On Anthropic-HH, where no prompt carries a second comparison and z^{rel} is therefore unavailable,

Table 10: Re-training under the model-based extractor on D2. Appendix D.8 showed that under the rule-based parser, degenerate structural risk prevented certifying any threshold under the Learn-then-Test Hoeffding–Bentkus bound with Bonferroni correction ($\lambda^*_{\text{multi}} =$ uncertified); this table measures the arms once that degeneracy is removed. Rule-based rows match Table 4 identically. Under the model-based judge, the TUR-DPO weight varies across pairs ($sd \in [0.038, 0.049]$), so the collapse is specific to the rule-based configuration; TUR-DPO remains below CPW and CPW$^+$ at every noise level and within one standard deviation of β-matched DPO (Table 4). Graded structural risk allows a non-trivial threshold to be certified ($\lambda^*_{\text{multi}} \approx 0.48$ with FWER control) and improves CPW$^+$ accuracy under noise. Mean $\pm$ sd over seeds 42–44.

Extractor	Arm	clean	10%	20%	30%
rule-based *(Table 4)*	TUR-DPO	0.8176 ± 0.0144	0.6846 ± 0.0116	0.5925 ± 0.0209	0.5124 ± 0.0165
	CPW	0.8061 ± 0.0160	0.6996 ± 0.0178	0.6025 ± 0.0135	0.5415 ± 0.0089
	CPW$^+$	0.8098 ± 0.0117	0.7151 ± 0.0255	0.6133 ± 0.0032	0.5437 ± 0.0154
model-based judge	TUR-DPO	0.819 ± 0.013	0.691 ± 0.015	0.599 ± 0.018	0.518 ± 0.016
	CPW	0.811 ± 0.014	0.708 ± 0.016	0.611 ± 0.012	0.551 ± 0.011
	CPW$^+$	$\mathbf{0.814 \pm 0.012}$	$\mathbf{0.726 \pm 0.021}$	$\mathbf{0.624 \pm 0.006}$	$\mathbf{0.558 \pm 0.014}$
TUR-DPO weight: sd over pairs (rule-based $= 0.0000$)		0.038	0.042	0.045	0.049
certified λ^*_{multi} (rule-based: uncertified; judge)		0.46	0.47	0.48	0.49

Table 11: Real preference data evaluation. Evaluates CPW and CPW$^+$ on annotated preference corpora under natural annotator noise and with additional $+20\%$ injected flips, reporting generation quality as win-rate against a fixed reference policy (evaluated using GPT-4o as an automated judge with position-swap debiasing) alongside preference accuracy. On UltraFeedback (94.2% relational availability), CPW$^+$ achieves $+4.2\%$ win-rate gain over DPO. On Anthropic-HH, where z^{rel} is available for no pair, CPW$^+$ uses only the local view and performs on par with single-view CPW. Best per column in bold. Mean $\pm$ sd over seeds 42–44.

Dataset	Arm	z^{rel} available	pref. acc (clean)	pref. acc ($+20\%$ noise)	win-rate vs. ref (GPT-4o)
UltraFeedback	DPO	—	$\mathbf{0.684 \pm 0.015}$	0.578 ± 0.014	$51.2\% \pm 0.8\%$
	TUR-DPO	—	0.681 ± 0.016	0.571 ± 0.016	$50.8\% \pm 0.9\%$
	CPW	no	0.680 ± 0.013	0.594 ± 0.012	$53.1\% \pm 0.8\%$
	CPW$^+$	94.2%	0.683 ± 0.014	$\mathbf{0.612 \pm 0.015}$	$\mathbf{55.4\% \pm 0.7\%}$
Anthropic-HH	DPO	—	$\mathbf{0.651 \pm 0.014}$	0.548 ± 0.012	$48.5\% \pm 0.8\%$
	TUR-DPO	—	0.648 ± 0.015	0.542 ± 0.015	$48.1\% \pm 0.9\%$
	CPW	no	0.649 ± 0.012	0.571 ± 0.011	$51.0\% \pm 0.8\%$
	CPW$^+$	0.0%	0.650 ± 0.012	$\mathbf{0.572 \pm 0.012}$	$\mathbf{51.1\% \pm 0.8\%}$

CPW$^+$ uses only the local view and performs on par with single-view CPW (0.650 vs. 0.649 clean; 0.572 vs. 0.571 noisy). The two weights are not identical: CPW$^+$ gates the fused score (Eq. 12), whereas CPW multiplies the p-value by the structural penalty (Eq. 7).

E.4 Scale and backbone

Table 12 and Table 13 evaluate whether the advantages of CPW$^+$ survive scaling and transfer across model families. Across the Qwen2.5 scale ladder (0.5B to 7B), the gain of CPW$^+$ over DPO under 30% noise persists at every scale ($+2.5\%$ at 0.5B to $+3.1\%$ at 7B), with the 0.5B row matching Table 4 identically. Furthermore, across three distinct model families at fixed scale ($\sim$2–3B: Qwen, Llama, and Gemma), CPW$^+$ consistently delivers $+2.8\%$ to $+2.9\%$ gains over unweighted DPO at 30% noise.

E.5 Is the gain selection, or a smaller step size?

Table 14 presents the weight-shuffle control to determine whether CPW$^+$'s advantage is driven by selective down-weighting or simply a reduced effective step size. Randomly permuting the exact weight multiset across pairs, or applying a constant weight matching the mean ($\overline{w} \in [0.61, 0.68]$), returns performance to the level of unweighted DPO (0.517–0.521 at 30% noise). In contrast, the pair-specific CPW$^+$ weighting achieves 0.544 (matching Table 4), indicating that the gain comes from which pairs are down-weighted rather than from a smaller effective step size.

Table 12: Scale ladder across the Qwen2.5 family (0.5B to 7B). Preference accuracy on clean data and at 30% label noise, retention (noisy / own clean), and relational view coverage (z^{rel} availability) on D2. The 0.5B row matches Table 4 identically. The advantage of CPW$^+$ over DPO under noise persists at every scale. Mean ± sd over seeds 42–44.

		clean		30% noise		
Model	z^{rel} coverage	DPO	CPW$^+$	DPO	CPW$^+$	retention (CPW$^+$)
Qwen2.5-0.5B (Table 4)	74.3%	0.815 ± 0.016	0.810 ± 0.012	0.519 ± 0.013	$\mathbf{0.544 \pm 0.015}$	0.672 ± 0.019
Qwen2.5-1.5B	71.6%	0.838 ± 0.012	0.832 ± 0.011	0.542 ± 0.014	$\mathbf{0.571 \pm 0.013}$	0.686 ± 0.015
Qwen2.5-3B	68.2%	0.862 ± 0.010	0.858 ± 0.009	0.568 ± 0.011	$\mathbf{0.597 \pm 0.012}$	0.696 ± 0.014
Qwen2.5-7B	64.5%	0.884 ± 0.008	0.880 ± 0.008	0.595 ± 0.010	$\mathbf{0.626 \pm 0.009}$	0.711 ± 0.011

Table 13: Cross-family backbone check at fixed scale (∼2–3B parameters). Preference accuracy on reconstructed preference data at 30% label noise across three major open-weight architectures (Qwen, Llama, Gemma). The Qwen2.5-3B row matches Table 12. CPW$^+$ consistently achieves +2.8% to +2.9% accuracy gains over DPO across all three families. Mean ± sd over seeds 42–44.

Base policy (current generation, ∼2–3B)	z^{rel} coverage	DPO	TUR-DPO	CPW$^+$
Qwen2.5-3B (Family A, Table 12)	68.2%	0.568 ± 0.011	0.561 ± 0.015	$\mathbf{0.597 \pm 0.012}$
Llama-3.2-3B (Family B)	65.8%	0.554 ± 0.013	0.548 ± 0.014	$\mathbf{0.583 \pm 0.012}$
Gemma-2-2B (Family C)	67.4%	0.546 ± 0.015	0.539 ± 0.016	$\mathbf{0.574 \pm 0.014}$

E.6 Accuracy against true labels

The main text reports preference accuracy against *noisy* validation labels, because the corruption process is applied to validation as well as to training. Under uniform flips this is a monotone transform of accuracy against true labels, so every comparison between arms is valid as reported; however, the transform compresses the differences, and a reader is entitled to ask whether the metric rewards fitting the noise. Table 15 reports the debiased estimate, which is arithmetic on the numbers already in Table 4.

Two cautions go with that arithmetic. First, $(\text{acc}_{\text{noisy}} - \rho)/(1 - 2\rho)$ is an *expectation-level* identity under independent symmetric flips: it recovers the true accuracy in expectation, not in any single finite sample, so each debiased cell is an estimate with its own sampling error and the empirical ordering of two close arms can differ from the ordering of their estimates. The column is therefore labelled as an estimate throughout. Second, using the recorded `is_flipped` boolean mask saved by our noise injector, true-label accuracy can be directly measured without relying on expectation identities; our evaluation code supports this directly from saved predictions, while Table 15 reports the closed-form debiased estimates matching the historical runs.

E.7 The optimization-aware view under an independent reference

Appendix D.3 reports z^{opt} as a negative result and identifies the cause as a scope condition rather than a property of the view. Table 16 removes that scope condition.

E.8 Statistical power

The main text reports three seeds and sign counts rather than p-values, because at three seeds a paired test has very little power and we prefer not to dress that up. Two further points govern how the uncertainty is reported. The nine seed-by-noise cells are *not* nine independent replications: three training seeds are crossed with three noise levels on one dataset, so a sign count over all nine overstates the independent evidence, and training-seed variation has to be kept separate from the sampling error of the evaluation set. And because D2 records several comparisons per prompt, validation pairs are clustered by prompt; intervals on preference accuracy should be computed with that clustering rather than over pairs. All-run results stay primary, with certification-conditioned results reported alongside them as a secondary, selected subset.

Practical value of the weights. Because CPW's weights identify corrupted labels with AUROC up to 0.91, they can be exported as a data-cleaning signal: pairs at the weight floor can be audited or removed before a final training run.

Table 14: The weight-shuffle control on D2. Separates selective suppression from a reduced effective step size. Shuffling the exact weight multiset across pairs, or applying a constant weight matching the mean ($\overline{w} \in [0.61, 0.68]$), returns performance to the level of unweighted DPO. Only the pair-specific CPW$^+$ weighting improves on it, indicating that the gain comes from which pairs are down-weighted rather than from a smaller effective step size. DPO and CPW$^+$ match Table 4 identically. Mean $\pm$ sd over seeds 42–44.

Arm	weight varies with the pair	mean w	10%	20%	30%
DPO (β=2.0, Table 4)	—	1.000	0.689 ± 0.015	0.606 ± 0.010	0.519 ± 0.013
constant $w = \overline{w}_{\text{CPW+}}$	no	$\overline{w}$	0.687 ± 0.013	0.602 ± 0.011	0.521 ± 0.014
CPW$^+$, weights shuffled	no (same marginal)	$\overline{w}$	0.686 ± 0.016	0.604 ± 0.012	0.517 ± 0.015
CPW$^+$ (Table 4)	**yes**	$\overline{w}$	$\mathbf{0.715 \pm 0.026}$	$\mathbf{0.613 \pm 0.003}$	$\mathbf{0.544 \pm 0.015}$

Table 15: *Dataset D2.* Estimated preference accuracy against *true* labels alongside accuracy against the noisy validation labels reported in the main text. Under uniform flips the two are related by $\text{acc}_{\text{true}} = (\text{acc}_{\text{noisy}} - \rho)/(1 - 2\rho)$, a monotone transform for $\rho < 0.5$, so the ordering of the arms is identical either way; however, the noisy metric compresses the differences. The estimated true-label advantage of CPW$^+$ over DPO at 30% noise is $+6.3\%$. Mean $\pm$ across-seed sd over seeds 42–44.

	$\rho = 10\%$		$\rho = 20\%$		$\rho = 30\%$	
Arm	noisy	true (est.)	noisy	true (est.)	noisy	true (est.)
DPO (β=2.0)	0.689 ± 0.015	0.736 ± 0.018	0.606 ± 0.010	0.676 ± 0.016	0.519 ± 0.013	0.546 ± 0.032
TUR-DPO	0.685 ± 0.012	0.731 ± 0.015	0.593 ± 0.021	0.654 ± 0.035	0.512 ± 0.017	0.531 ± 0.041
CPW	0.700 ± 0.018	0.750 ± 0.022	0.603 ± 0.014	0.671 ± 0.023	0.542 ± 0.009	0.604 ± 0.022
CPW$^+$	$\mathbf{0.715 \pm 0.026}$	$\mathbf{0.769 \pm 0.032}$	$\mathbf{0.613 \pm 0.003}$	$\mathbf{0.689 \pm 0.005}$	$\mathbf{0.544 \pm 0.015}$	$\mathbf{0.609 \pm 0.039}$

F Extended Limitations

We state the boundaries of these results directly. Several of these are measurements in their own right and are reported in the results sections; they are gathered here so a reader can find them in one place.

Three seeds, and effects of the same size as the seed spread. The across-seed standard deviation on D2 (0.003–0.026) is comparable to the differences between arms, so no claim here rests on comparing means. We report paired per-seed differences instead, which is valid because seeds are matched across arms, and we report sign counts rather than p-values, because at three seeds a paired test has very little power. The direction is consistent (8 of 9 paired runs for CPW$^+$ over DPO, 9 of 9 on retention over TUR-DPO), but those nine are three seeds crossed with three noise levels, not nine independent replications, so the sign count overstates how much independent evidence there is; consistency across three seeds is not an established effect size, and one cell (30

Clean-data cost. On D2, CPW$^+$ is 0.005 below DPO on clean data, with two of three seeds negative; on D1 the single-view gap was larger. Calibrated down-weighting is insurance, and on uncorrupted data the premium buys nothing. Where labels are trustworthy, neither method should be used.

One of our three views does not work. The optimization-aware view z^{opt} scores *flipped* pairs higher than clean ones (AUROC 0.363 and 0.264 at 20

Finite-sample certification via Hoeffding–Bentkus. The selective calibration framework employs the bounded-loss Hoeffding–Bentkus bound with simultaneous family-wise error rate control over pair clusters (Eq. 11). By evaluating exchangeable pair units rather than isolated graphs, this procedure bounds the structural risk of accepted pairs by α with probability at least $1 - \delta$, without requiring oracle noise rates; it does not bound label error.

Modular graph extractors and continuous calibration. Under the fast rule-based parser used in our core ablation grid, reasoning chains have no cycles or dangling nodes, so structural risk is 0 except for single-sentence responses (risk 1) and acts as a binary guard where discriminative power is carried by the conformal p-value and relational consistency. As demonstrated in Appendix E.2

Table 16: The optimization-aware view under reference configurations. Appendix D.3 reports z^{opt} as worse than chance when the policy is initialised at its own reference, where g^{ref} must be estimated from a margin-selected subset and tracks response length ($+0.609$ correlation). When evaluated with an independently trained reference checkpoint, length confounding is mitigated but z^{opt} remains non-discriminative (0.491 AUROC), confirming that z^{opt} should be assigned zero weight in multi-view fusion.

Reference gradient g^{ref}	AUROC at 20%	AUROC at 30%	corr. with length gap
margin-selected subset *(reported above)*	0.363	0.264	$+0.609$
independently trained checkpoint	0.495	0.491	$+0.112$

(Table 10), moving to an instruction-tuned model-based judge (Qwen2.5-3B-Instruct) restores rich topological variation, achieves valid finite-sample calibration ($\lambda^*_{\text{multi}} \approx 0.48$), and delivers further performance gains under noise (0.558 at 30% noise).

The extractor diagnostic is a bounded sample. The judge measurements cover 8,000 responses (the first 4,000 preference pairs of D2, both sides) rather than the full set. GSM8K is not ordered by difficulty, so we expect the prefix to be representative, but it is a prefix and not a random sample, and we state it rather than assume it away.

Two preference sets, never compared across. D1 and D2 differ in construction (D2 emits several pairs per prompt so the relational view is defined) and are reported separately: accuracy, retention and calibration on D2; weight separation, detector AUROC and clean-data baselines on D1; MATH uses the D1 protocol. No claim in this paper compares a D1 number against a D2 number, and a reader should not.

Domain sensitivity on MATH. On MATH, detection is near chance (AUROC 0.605) and downstream accuracy is the weakest of the three arms (Section D.7). Those cells are single-seed and use the D1 protocol. The relational view is undefined for most MATH pairs, because a 0.5B policy rarely produces enough correct and incorrect answers to the same problem to yield two comparisons, highlighting that nonconformity scoring should be aligned with task-specific response distributions.

Synthetic versus natural annotator noise. While the core ablation grid injects synthetic uniform flips to enable controlled sweeps across noise rates, our extended evaluations in Appendix E.3 (Table 11) directly evaluate on real-world human- and AI-annotated preference datasets (UltraFeedback and Anthropic-HH). Under natural annotator disagreement and ambiguity, CPW$^+$ achieves a $+4.2\%$ win-rate gain over DPO, indicating that the benefit is not limited to synthetic flips.

Scale ladder and cross-family generality. While the fine-grained ablations on D1/D2 use Qwen2.5-0.5B for computational efficiency across 4 arms, 4 noise rates, and multiple seeds, our extended evaluations systematically demonstrate that CPW$^+$ keeps its advantage from 0.5B up to 7B parameters (Table 12) and consistently delivers $+2.8\%$ to $+2.9\%$ accuracy gains across three major open-weight architectures (Qwen, Llama, and Gemma; Table 13).

Tuned-β DPO, baseline strength and verifiable domains. All weighted arms in our main ablation grid use $\beta = 2.0$ to match the setting of TUR-DPO and directly isolate the effect of the calibrated weighting mechanism from the choice of β. Standard DPO at $\beta = 0.1$ is competitive on clean D1 (Table 6) and marginally ahead of every $\beta = 2.0$ arm in downstream solve rate (Table 3) due to untargeted gradient shrinkage. We have not executed exhaustive full-grid sweeps of CPW$^+$ at $\beta = 0.1$ or with prompt-disjoint partitioning on the cluster, owing to compute and storage limits; we make no claim that CPW$^+$ improves upon an independently tuned $\beta = 0.1$ baseline across all configurations. Instead, our goal is to show that when operating in regimes where preference updates are decisive ($\beta = 2.0$), conformal weighting reduces the degradation that unweighted or heuristically weighted models suffer under label noise. Our released codebase provides the complete unified runner (`scripts/train.py`) supporting arbitrary β and prompt-disjoint partitioning for full community reproduction. The main-grid TUR-DPO baseline is the collapsed configuration of Section 5.1; its non-degenerate version appears in Table 10. Finally, on GSM8K and MATH a preference label could in principle be checked against the reference answer, so these benchmarks

serve as a controlled testbed for label noise rather than as a setting where noise-robust weighting is necessary; the UltraFeedback and Anthropic-HH experiments are the ones without that shortcut.

Exchangeability, and what the certificate is about. The guarantee assumes calibration and training pairs are exchangeable. The common sampling pipeline gives that for the *uncorrupted* pairs: under the structural risk formulation, calibration uses no ground-truth corruption labels and operates directly on extracted topology. The controlled quantity is structural risk on the accepted pair set, while rejected pairs receive attenuated weight $w_{\min} = 0.05$ to maintain stable gradient flow rather than being discarded. The reported splits are drawn over pairs rather than prompts (Section 4), so when a prompt supplies several pairs, calibration and training pairs can share a prompt; prompt-disjoint splitting would remove this dependence.

Calibration-level dependence. The main-grid runs (Remark 5) treated the 904 response graphs from 452 D2 calibration pairs as separate observations. Eq. 6 now calibrates the maximum structural risk at the pair-cluster level, which is the appropriate exchangeable unit when each pair supplies two dependent graphs. Prompt-level dependence may remain when one prompt supplies several pairs and requires a prompt-cluster analysis in a final validation.