

Reward Modeling for Zen Alignment: Multi-Reward Ensembles and Synthetic Preference Generation

Technical Report v2025.04

Zen LM Research Team
research@zenlm.org

April 2025

Abstract

Reward modeling is a critical component of aligning large language models to human preferences. We present the Zen alignment stack, comprising: (1) a preference data collection pipeline that produces high-quality human judgments at scale; (2) ensemble reward models based on the Bradley-Terry statistical model with multi-head architecture; (3) synthetic preference generation using a red-team/blue-team approach to augment human data; and (4) Direct Preference Optimization (DPO) as an alternative to full RL training. Our multi-reward ensemble achieves 87.3% correlation with human judgment on a held-out evaluation set, a 12% improvement over single-model baselines. RLHF with our ensemble reward substantially outperforms DPO alone on open-ended generation quality, while DPO achieves competitive results at lower compute cost for instruction following.

Contents

1	Introduction	3
1.1	Alignment Taxonomy	3
2	Preference Data Collection	3
2.1	Annotation Protocol	3
2.2	Inter-Annotator Agreement	3
2.3	Dataset Composition	4
3	Bradley-Terry Reward Model	4
3.1	Bradley-Terry Model	4
3.2	Multi-Head Architecture	4
3.3	Ensemble Construction	5
4	Synthetic Preference Generation	5
4.1	Red-Team / Blue-Team Protocol	5
4.2	Quality Filtering	5
5	Direct Preference Optimization (DPO)	5
6	Experiments	6
6.1	Reward Model Accuracy	6
6.2	RLHF vs. DPO Comparison	6
6.3	Reward Model Calibration	6

6.4	Downstream RLHF Results	6
6.5	Reward Hacking Analysis	7
7	Discussion	7
7.1	Ensemble vs. Single Reward Model	7
7.2	Synthetic Data Quality	7
7.3	DPO Compute-Quality Trade-off	7
8	Conclusion	7

1 Introduction

Training a large language model to generate text that humans find helpful, harmless, and honest (HHH) [1] requires a reward signal that accurately captures human preferences. Reinforcement Learning from Human Feedback (RLHF) [2, 3] trains a reward model on preference data and then uses it to fine-tune the policy via PPO. The quality of the reward model is a critical bottleneck: a poorly calibrated reward model produces *reward hacking*, where the policy exploits reward model weaknesses rather than genuinely improving.

We address reward model quality through three mechanisms:

1. **High-quality preference data:** structured collection of pairwise human judgments with inter-annotator agreement metrics.
2. **Multi-reward ensemble:** an ensemble of reward models with different training objectives (helpfulness, harmlessness, honesty, instruction-following), combined to produce a holistic alignment score.
3. **Synthetic preference generation:** using a red-team model to generate challenging response pairs that expose alignment failures, augmenting human data with synthetically generated hard cases.

1.1 Alignment Taxonomy

Following [1], we distinguish:

- **Helpfulness:** whether the response addresses the user’s intent effectively.
- **Harmlessness:** whether the response avoids harmful content or enabling harm.
- **Honesty:** whether the response is factually accurate and appropriately uncertain.
- **Instruction following:** whether the response adheres to specified format, length, and style constraints.

These dimensions are partially correlated but not identical, motivating a multi-head reward architecture.

2 Preference Data Collection

2.1 Annotation Protocol

Human annotators compare pairs of responses (y_A, y_B) to the same prompt x and indicate: strongly prefer A , slightly prefer A , about equal, slightly prefer B , strongly prefer B . This 5-point scale is mapped to a continuous preference signal $\delta \in \{-2, -1, 0, +1, +2\}$ for the Bradley-Terry likelihood computation.

Annotators also rate each response on four dimensions (helpfulness, harmlessness, honesty, instruction-following) using a 1–5 Likert scale. These dimensional ratings train the individual reward model heads.

2.2 Inter-Annotator Agreement

To ensure data quality, each prompt-pair is annotated by 3 independent annotators. We compute Fleiss’s κ as a quality metric:

Table 1: Inter-annotator agreement across preference data categories.

Category	Fleiss’s κ	Agreement interpretation
Helpfulness	0.68	Substantial
Harmlessness	0.81	Almost perfect
Honesty	0.71	Substantial
Instruction following	0.87	Almost perfect
Overall preference	0.64	Substantial

Prompt-pairs with $\kappa < 0.4$ are excluded from training, retaining 89% of collected pairs. This yields a final dataset of 1.2M preference pairs.

2.3 Dataset Composition

Table 2: Preference dataset composition by domain.

Domain	Pairs (K)	Fraction (%)	Avg. κ
General conversation	320	26.7	0.63
Instruction following	210	17.5	0.88
Mathematics / reasoning	180	15.0	0.82
Code generation	150	12.5	0.79
Factual QA	140	11.7	0.71
Safety-critical	120	10.0	0.84
Creative writing	80	6.7	0.58
Total	1200	100.0	0.74

3 Bradley-Terry Reward Model

3.1 Bradley-Terry Model

The Bradley-Terry (BT) model [4] models the probability that response y_A is preferred over y_B for prompt x as:

$$P(y_A \succ y_B \mid x) = \sigma(r(x, y_A) - r(x, y_B)) \quad (1)$$

where $r(x, y) \in \mathbb{R}$ is the reward score and σ is the sigmoid function. The reward model is trained to maximize the log-likelihood:

$$\mathcal{L}_{\text{BT}} = -\mathbb{E}_{(x, y_A, y_B, \delta) \sim \mathcal{D}} [\delta \log P(y_A \succ y_B \mid x)] \quad (2)$$

where $\delta \in \{-2, -1, 0, +1, +2\}$ is the annotator preference strength.

3.2 Multi-Head Architecture

Our reward model uses a Zen MoDE encoder to produce a contextual representation of (x, y) , with separate output heads for each alignment dimension:

$$r_d(x, y) = \mathbf{w}_d^\top h(x, y) + b_d, \quad d \in \{\text{help, harm, hon, inst}\} \quad (3)$$

where $h(x, y) \in \mathbb{R}^{d_{\text{model}}}$ is the [EOS] token representation from the final transformer layer. The combined reward is:

$$r(x, y) = \sum_d \lambda_d \cdot r_d(x, y) \quad (4)$$

with per-domain weights λ_d learned on the validation set.

3.3 Ensemble Construction

We train 5 reward models with different random seeds and data orderings, then ensemble:

$$r_{\text{ens}}(x, y) = \frac{1}{5} \sum_{k=1}^5 r^{(k)}(x, y) \quad (5)$$

Ensemble disagreement $\text{Var}_k[r^{(k)}(x, y)]$ serves as an uncertainty estimate: high-variance examples are flagged for additional human review during RLHF training.

4 Synthetic Preference Generation

4.1 Red-Team / Blue-Team Protocol

Human preference data is expensive and limited in coverage. We augment it using a red-team model trained to generate challenging prompts that expose alignment failures, and a blue-team (the base policy) that responds. The red-team generates prompts by:

1. Sampling a seed prompt x_0 from the training distribution.
2. Applying a perturbation Δ that targets known weak spots (jailbreaks, out-of-distribution topics, adversarial formatting).
3. Generating the perturbed prompt $x' = T(x_0, \Delta)$.

For each perturbed prompt, the base policy generates two responses with different decoding temperatures ($T_1 = 0.7$, $T_2 = 1.2$), producing a natural diversity in response quality. The reward model ensemble scores both responses and the pair is added to the preference dataset with the model scores as soft labels.

4.2 Quality Filtering

Synthetic pairs are filtered by:

- Ensemble agreement: $\text{Var}_k[r^{(k)}] < \sigma_{\text{thresh}}$ ensures the preference label is well-defined.
- Margin threshold: $|r_{\text{ens}}(y_A) - r_{\text{ens}}(y_B)| > m_{\text{min}}$ ensures the pair is informative (non-trivial).
- Semantic diversity: the prompt x' must differ from training prompts by at least 0.15 in embedding distance.

This process generates 8M synthetic preference pairs, of which 3.4M pass all filters.

5 Direct Preference Optimization (DPO)

DPO [5] directly optimizes the policy using preference data, bypassing explicit reward model training:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x, y_w, y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w | x)}{\pi_{\text{ref}}(y_w | x)} - \beta \log \frac{\pi_{\theta}(y_l | x)}{\pi_{\text{ref}}(y_l | x)} \right) \right] \quad (6)$$

where y_w and y_l are preferred and dispreferred responses, π_{ref} is the reference (SFT) policy, and β controls the KL penalty.

We compare DPO trained on our human + synthetic preference data to full RLHF using PPO with our ensemble reward model.

6 Experiments

6.1 Reward Model Accuracy

Table 3: Reward model accuracy on held-out human preference pairs (binary: correct if reward model prefers the human-preferred response).

Model	Accuracy (%)	Human corr. (Pearson r)
Single RM (no ensemble)	77.8	0.778
Ensemble (5 RMs)	81.4	0.814
Multi-head + ensemble	85.2	0.852
+ Synthetic data	87.3	0.873

6.2 RLHF vs. DPO Comparison

Table 4: Alignment benchmark results. AlpacaEval 2.0 win rate vs. GPT-4 Turbo. MT-Bench score (1–10). Arena Elo (Chatbot Arena).

Training	AlpacaEval 2.0 (%)	MT-Bench	Arena Elo
SFT only	18.4	7.8	1124
DPO (human data)	31.7	8.4	1198
DPO (+ synthetic)	36.2	8.7	1231
RLHF (PPO, single RM)	38.4	8.9	1248
RLHF (PPO, ensemble RM)	44.8	9.2	1312

Full RLHF with ensemble reward outperforms DPO by 8.6 pp on AlpacaEval. However, DPO at significantly lower compute cost achieves competitive performance on MT-Bench.

6.3 Reward Model Calibration

Table 5: ECE of reward model scores. Lower is better. The ensemble reduces miscalibration by 41% vs. single RM.

Model	ECE	Δ vs. single
Single RM	0.124	—
Ensemble (5 RMs)	0.082	−33.9%
Multi-head + ensemble	0.073	−41.1%

6.4 Downstream RLHF Results

Table 6: Downstream benchmark performance after RLHF. RLHF with ensemble RM provides consistent improvements without degrading base capabilities.

Model	MMLU	GSM8K	HumanEval	TruthfulQA
Zen MoDE-72B (base)	84.7	90.8	79.3	74.8
+ SFT	84.9	91.2	80.1	76.3
+ DPO	85.1	91.8	80.7	78.9
+ RLHF (ensemble RM)	85.4	92.1	81.3	82.4

6.5 Reward Hacking Analysis

Table 7: Reward hacking incidence: fraction of RLHF checkpoints where reward model score increases but human preference decreases.

Configuration	Hacking rate (%)	Detected by ensemble (%)
Single RM, no detection	14.2	—
Ensemble RM	7.8	89.3% of single-RM hacking events
Ensemble + variance gate	3.1	97.2% of single-RM hacking events

The variance gate (flagging high-disagreement ensemble predictions) detects 97% of reward hacking events at the cost of rejecting 8% of legitimate training updates.

7 Discussion

7.1 Ensemble vs. Single Reward Model

The ensemble provides two benefits beyond accuracy: (1) calibration improvement via variance-based uncertainty estimation, and (2) reward hacking detection through disagreement signals. The computational overhead of running 5 reward models during RLHF rollouts is $5\times$ for inference, but reward model inference is typically $<10\%$ of total PPO compute, making this tractable.

7.2 Synthetic Data Quality

The synthetic preference pairs improve accuracy by 2.1 pp (Table 3). Quality filtering is critical: unfiltered synthetic data degrades accuracy by 1.4 pp due to noisy margin pairs. The margin threshold $m_{\min}$ is the most important filter.

7.3 DPO Compute-Quality Trade-off

DPO avoids reward model training and PPO rollout generation, reducing alignment compute by approximately $8\times$. For deployment scenarios where compute is the primary constraint, DPO with synthetic preference augmentation is the recommended choice, achieving 81% of full RLHF performance at 12.5% of the compute.

8 Conclusion

We have presented the Zen alignment stack: multi-head ensemble reward models trained on 1.2M human preferences plus 3.4M filtered synthetic pairs, combined with RLHF or DPO fine-tuning. Key results: 87.3% reward model human correlation, 44.8% AlpacaEval win rate after RLHF, and 97% reward hacking detection with ensemble variance gating. The full stack is recommended for production deployments; DPO is recommended when compute is constrained.

References

- [1] A. Askell, Y. Bai, A. Chen, et al. *A General Language Assistant as a Laboratory for Alignment*. arXiv:2112.00861, 2021.
- [2] P. Christiano, J. Leike, T. Brown, et al. *Deep Reinforcement Learning from Human Preferences*. NeurIPS, 2017.

- [3] L. Ouyang, J. Wu, X. Jiang, et al. *Training Language Models to Follow Instructions with Human Feedback*. NeurIPS, 2022.
- [4] R.A. Bradley, M.E. Terry. *Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons*. Biometrika, 39(3/4):324–345, 1952.
- [5] R. Rafailov, A. Sharma, E. Mitchell, et al. *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. NeurIPS, 2023.