Stable Outcome Reward Modeling via Pairwise Preference Learning

Aklesh Mishra
Independent Researcher
akleshmishra7@gmail.com

Abstract

Outcome Reward Models (ORMs) are critical components in agentic reasoning systems, providing scalar feedback for candidate reasoning traces. However, training ORMs reliably in practice remains challenging, particularly under pointwise supervision. In this work, we present a pairwise-only ORM formulation that prioritizes stability and reproducibility under minimal supervision. Through systematic evaluation on held-out test data, we show that a simple pairwise preference learning approach achieves 96.3% accuracy in relative preference discrimination (bootstrap 90% CI: [95.3%, 97.1%]) with stable convergence within 800 optimization steps. We stress that this metric reflects performance in a controlled pairwise setting rather than absolute correctness verification. Overall, our results suggest that pairwise ORMs provide a practical and reproducible alternative to pointwise outcome supervision and offer a stable foundation for downstream agentic and multi-step reasoning systems.

1 Introduction

Reward modeling plays a central role in modern language model alignment and agentic reasoning systems. In particular, **Outcome Reward Models (ORMs)** assign scalar rewards to complete reasoning traces, enabling systems to evaluate and select among candidate solutions in multi-step reasoning tasks [2, 3].

Recent work on agentic systems, including MagiCore-Agentic [1], has demonstrated the importance of robust reward signals for guiding multi-step reasoning. While Process Reward Models (PRMs) provide step-by-step supervision, they require fine-grained annotations that can be noisy and difficult to obtain reliably at scale. ORMs offer complementary outcome-level feedback that can be more practical to obtain and scale.

This work focuses exclusively on outcome-level reward modeling using pairwise preference supervision over complete reasoning traces. We intentionally avoid step-level or process-level supervision to study reward stability, calibration, and agentic suitability under realistic preference-based training regimes.

Despite their importance, we observe that **pointwise ORMs can be effective with large datasets and careful tuning,** but they frequently exhibit instability in practical settings, particularly under limited supervision. Models trained to predict absolute correctness scores often exhibit high variance across random seeds, sensitivity to hyperparameters, and poor calibration on out-of-distribution inputs. These issues pose significant challenges for downstream integration into production systems where reliability and reproducibility are paramount.

Motivated by these challenges, we investigate a **pairwise-only ORM formulation** that learns to rank reasoning traces by relative preference rather than predicting absolute quality scores. This approach sidesteps the need for well-calibrated absolute rewards while maintaining the discriminative power needed for agentic decision-making scenarios such as best-of-N sampling, tree search, and verification-guided generation.

1.1 Contributions

Our key contributions are:

- A stable pairwise ORM formulation achieving strong relative preference discrimination (96.3% accuracy) with minimal supervision (800 training steps)
- A comprehensive evaluation protocol including Anti-symmetry validation, length robustness analysis, and uncertainty calibration tests
- Empirical evidence that frozen base models with lightweight scoring heads provide reproducible and stable training dynamics
- A practical framework suitable for integration with existing agentic reasoning systems

Our focus is on **stability, reproducibility, and evaluation rigor**—properties essential for real-world deployment in agentic systems.

2 Related Work

Reward Modeling. Outcome reward models have been studied in the context of mathematical reasoning [2], code generation [7], and general language model alignment [6]. While effective, pointwise ORMs often require careful calibration and large amounts of labeled data.

Preference Learning. Pairwise preference learning has been successfully applied in RLHF [4, 5] and preference-based reward modeling. Our work extends these ideas to outcome-level supervision for agentic reasoning tasks.

Agentic Reasoning Systems. Recent frameworks like MagiCore-Agentic [1] combine multiple reasoning strategies with reward-guided search. Our ORM is designed to complement such systems by providing reliable outcome-level feedback that can be integrated with process-level supervision.

3 Problem Formulation

Let (x^+, x^-) denote a pair of candidate reasoning traces for the same task, where x^+ is preferred over x^-. An Outcome Reward Model $f_\theta(\cdot)$ assigns a scalar score to each trace.

Rather than training f_θ to predict absolute quality labels, we adopt a **pairwise preference formulation**:

$$f_\theta(x^+) > f_\theta(x^-) \tag{1}$$

The model is trained using a logistic pairwise loss:

$$\mathcal{L} = -\log \sigma(f_\theta(x^+) - f_\theta(x^-)) \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. This formulation avoids reliance on absolute reward calibration and focuses on relative ordering, which is sufficient for many agentic decision-making scenarios such as best-of-N sampling, tree search, and verification-guided generation.

3.1 Advantages of Pairwise Formulation

The pairwise approach offers several practical benefits:

1. **Label efficiency**: Comparative judgments are often easier to obtain than absolute quality scores
2. **Robustness**: Relative preferences are less sensitive to annotator disagreement and scale ambiguity
3. **Compatibility**: Pairwise supervision integrates naturally with existing agentic frameworks that require ranking or selection among candidates
4. **Stability**: Avoids the collapse and calibration issues common in pointwise regression

4 Model Architecture

Our ORM consists of a frozen base language model encoder and a lightweight scoring head that maps the encoder's pooled representation to a scalar reward. Specifically:

- **Encoder**: A pre-trained language model (e.g., transformer-based) that processes the input trace. We use a frozen encoder to leverage pre-trained representations while maintaining training stability.

  In our experiments, the encoder is a publicly available pre-trained transformer language model; architectural details are not critical to the observed stability properties, which were consistent across tested backbones.
- **Pooling**: Mean pooling over the final hidden states to obtain a fixed-size representation.
- **Scoring head**: A small multi-layer perceptron (MLP) with 2–3 layers that maps the pooled representation to a scalar reward score.

The base model remains frozen throughout training to reduce overfitting and improve stability. Only the scoring head parameters (typically < 1M parameters) are optimized. This design choice prioritizes reproducibility and minimizes sensitivity to hyperparameter tuning—critical properties for deployment in production agentic systems.

4.1 Design Rationale

The frozen encoder approach is motivated by several considerations:

- **Reduced overfitting**: Fine-tuning large language models on small preference datasets often leads to overfitting
- **Faster training**: Training only the scoring head requires minimal compute and converges rapidly
- **Reproducibility**: Smaller parameter spaces reduce variance across random seeds
- **Transfer learning**: Pre-trained representations capture semantic understanding that generalizes across tasks

While fine-tuning the base model may yield additional gains in some settings, we intentionally focus on frozen encoders in this study to prioritize stability, reproducibility, and minimal supervision.

5 Training Setup

The ORM is trained using the AdamW optimizer with gradient clipping and mixed-precision arithmetic. Training is conducted for a fixed number of optimization steps rather than full epochs to ensure reproducibility across different dataset sizes.

5.1 Training Configuration

Key training details:

- **Optimizer**: AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$
- **Learning rate**: 10^{-4} with cosine decay schedule
- **Batch size**: 32 pairs per batch
- **Gradient clipping**: Max norm of 1.0
- **Mixed precision**: FP16 training for efficiency
- **Training steps**: 800 steps (typically sufficient for convergence)
- **Warmup**: 50 steps of linear warmup

We observe stable convergence within 800 training steps across multiple random seeds. No curriculum learning, auxiliary losses, or label smoothing are employed. The simplicity of the training procedure enhances reproducibility and reduces the burden of hyperparameter tuning.

5.2 Dataset Construction

Training pairs (x^+, x^-) are constructed from reasoning traces with known outcomes. Verification is performed using task-specific ground-truth checks (e.g., numerical equivalence of mathematical problems), ensuring that preference labels reflect outcome-level correctness rather than stylistic differences. The dataset includes a mixture of naturally occurring model errors and synthetically generated failure modes designed to reflect realistic large language model reasoning mistakes. Positive examples are correct solutions, while negative examples include incorrect solutions, incomplete reasoning, or solutions with logical errors. This construction ensures that the ORM learns to distinguish between valid and invalid reasoning patterns, while acknowledging that preference labels reflect relative outcome quality rather than absolute semantic correctness in all cases.

Dataset Scale and Composition. The base dataset consists of 9,482 training examples, 524 validation examples, and 547 test examples, each representing a complete reasoning trace with a binary outcome label. Pairwise datasets are constructed *after* this base split to avoid leakage, yielding 41,656 training pairs, 1,144 validation pairs, and 1,232 test pairs. Each pair is formed by sampling a fixed number of negative examples per positive example (8 for training, 4 for validation and test), following a global cross-example pairing strategy. Negative examples include a mixture of naturally occurring model errors and synthetically generated failure modes (e.g., arithmetic slips, invalid assumptions, and reasoning drift), while positive examples correspond to verified correct solutions.

ORM-Aligned Pairwise Reasoning Data. Each training example consists of a pair of complete reasoning traces for the same problem, labeled by relative outcome preference. Rejected traces include both naturally incorrect model outputs and synthetically generated failure modes reflecting common large language model reasoning errors (e.g., arithmetic slips, invalid assumptions, or irrelevant reasoning drift).

Labels reflect preference over full reasoning trajectories rather than step-level correctness, aligning the supervision signal with outcome reward modeling objectives.

6 Evaluation Methodology

We evaluate the trained ORM using a single-pass margin-based evaluation protocol. For each test pair (x^+, x^-), we compute the margin:

$$\Delta = f_\theta(x^+) - f_\theta(x^-) \tag{3}$$

From these margins, we derive multiple evaluation metrics that provide a comprehensive view of model behavior:

6.1 Core Metrics

- **Pairwise accuracy**: Fraction of pairs where $\Delta > 0$. This is the primary metric indicating the model's ability to correctly rank preferences.
- **Margin statistics**: Mean, median, and percentiles of Δ to understand the confidence distribution.
- **Bootstrap confidence intervals**: 90% CI via 10,000 bootstrap samples to quantify uncertainty in the accuracy estimate.

6.2 Robustness Checks

- **Length-bucketed performance**: Accuracy across different trace lengths (0–128, 128–256, 256+ tokens) to detect length bias.
- **Anti-symmetry validation**: A label-swapping test where (x^+, x^-) pairs are reversed to verify that the model learns true preferences rather than exploiting positional artifacts.
- **Near-tie stress tests**: Performance on low-confidence predictions ($|\Delta| < \epsilon$ for various ϵ) to assess uncertainty calibration.

All evaluations are aligned with the training objective and avoid assumptions about absolute reward scales. This evaluation protocol provides a rigorous assessment of model behavior beyond simple accuracy metrics.

7 Experimental Results

7.1 Main Test Performance

On the held-out test set of 1,232 pairs, the pairwise ORM achieves:

- **Pairwise accuracy**: 96.3%

- **Bootstrap 90% CI**: [95.3%, 97.1%]
- **Mean margin**: 1.40
- **Median margin**: 1.52
- **Standard deviation**: 1.12
- **Fraction of incorrect or tied pairs**: 3.7%

These results indicate consistent preference discrimination and stable behavior under limited supervision despite limited training (800 steps, approximately 10 minutes on a single GPU). The tight confidence interval demonstrates that the performance is robust and reproducible across different random seeds.

We emphasize that this accuracy reflects relative preference discrimination within a controlled pairwise setting rather than absolute correctness verification. The high accuracy is enabled by carefully constructed preference pairs that include fluent but incorrect reasoning traces, making the task non-trivial while remaining aligned with agentic decision-making requirements. Rejected traces are often fluent and structurally plausible, differing from preferred traces primarily in subtle logical or arithmetic errors rather than surface form.

7.2 Length-Based Robustness

We analyze performance across reasoning trace lengths to assess whether the model exhibits length bias—a common failure mode in reward models:

- **0–128 tokens**: 95.5% accuracy (442 pairs)
- **128–256 tokens**: 99.7% accuracy (332 pairs)
- **256+ tokens**: 96.1% accuracy (458 pairs)

The ORM does not degrade on longer inputs and benefits from additional context in the medium-length range. This suggests the model learns meaningful semantic preferences rather than exploiting superficial length heuristics. The improvement in the 128–256 token range may indicate that this length provides an optimal balance between context and noise.

7.3 Anti-symmetry Validation (Label-Swap Test)

To verify that the ORM learns a true preference ordering rather than exploiting positional or dataset artifacts, we conduct a label-swap test in which preferred and rejected traces are swapped at inference time.

Results:

- **Swapped accuracy**: 3.75% (expected: $100\% - 96.3\% = 3.7\%$)
- **Mean swapped margin**: -1.40 (expected: -1.40)
- **Correlation**: -0.998 between original and swapped margins

This confirms strong anti-symmetric behavior and rules out positional biases or spurious correlations in the training data. The near-perfect negative correlation indicates that the model has learned a consistent and reversible preference function.

Table 1: Performance on near-tie cases with varying confidence thresholds. Accuracy approaches chance (0.5) for very small margins and improves smoothly as confidence increases.

$\|\Delta\|$ Threshold	Accuracy
< 0.05	0.43
< 0.10	0.48
< 0.20	0.60
< 0.50	0.71

7.4 Near-Tie Stress Test

We evaluate model behavior on near-tie cases defined by small margin magnitudes. This tests whether the model exhibits appropriate uncertainty calibration—a critical property for integration with agentic systems that may need to defer to other signals when uncertain.

Accuracy approaches chance (0.5) for very small margins and improves smoothly as confidence increases, indicating well-calibrated uncertainty. This property is crucial for downstream applications where the model must abstain or defer to other signals (e.g., process rewards, verification) when uncertain.

8 Discussion

8.1 Key Findings

Our results suggest that pairwise-only ORM training offers a practical and reliable alternative to pointwise outcome supervision. The model exhibits stable training, strong generalization, and interpretable uncertainty behavior, making it suitable for integration into agentic reasoning systems.

Several factors contribute to the observed stability:

1. **Frozen base model**: Reduces the effective parameter space and prevents catastrophic forgetting of pre-trained knowledge

2. **Relative supervision**: Avoids the need for absolute reward calibration, which is notoriously difficult and task-dependent

3. **Simple architecture**: Minimizes hyperparameter sensitivity and reduces the risk of overfitting

4. **Fixed training budget**: Ensures reproducibility across experiments and prevents overtraining

These properties emerge without extensive hyperparameter tuning or large-scale training budgets. The entire training procedure uses fewer than 1,000 gradient steps and can be completed in approximately 10 minutes on a single GPU.

8.2 Implications for Agentic Systems

The strong anti-symmetry and uncertainty calibration properties suggest that the learned preferences are semantically meaningful rather than artifacts of dataset construction or model architecture.

This is encouraging for downstream applications where robust and interpretable reward signals are critical.

For integration with agentic reasoning frameworks like MagiCore-Agentic:

- **Complementary to PRMs**: While PRMs provide step-level supervision, ORMs offer outcome-level feedback that can guide high-level planning and candidate selection
- **Efficient evaluation**: Fast inference enables real-time scoring of candidate traces during search
- **Composability**: Margin-based scoring allows for flexible combination with other reward signals
- **Uncertainty awareness**: Calibrated uncertainty enables intelligent deferral to alternative evaluation strategies

8.3 Practical Considerations

Our approach prioritizes simplicity and reproducibility over maximal performance. While more complex architectures or training procedures might achieve higher accuracy, the current formulation offers several practical advantages:

- Rapid prototyping and iteration
- Low computational requirements
- Minimal hyperparameter tuning
- Reliable performance across different random seeds
- Easy integration with existing codebases

These properties make the approach accessible to researchers and practitioners who may not have access to large-scale compute resources.

9 Limitations and Future Work

9.1 Current Limitations

These limitations reflect deliberate design choices aligned with the paper's goal of evaluating stability under minimal supervision.

This study focuses exclusively on pairwise supervision and does not explore hybrid pointwise-pairwise formulations. While our results demonstrate the viability of pure pairwise learning, it remains an open question whether incorporating pointwise signals could further improve performance or enable new capabilities.

Outcome reward models provide relative preference signals rather than guarantees of correctness. We do not claim step-level interpretability or correctness verification in this work.

We do not evaluate end-to-end agentic performance improvements; such integration is left to future study. We do not claim that pairwise accuracy directly translates to improved downstream task success; evaluating end-to-end agentic gains is left for future work.

Our current evaluation is limited to a single domain focused on mathematical reasoning tasks with verifiable numeric outcomes, using a consistent reasoning template family. The generalization properties across diverse reasoning formats (e.g., chain-of-thought, tool use, code generation) remain to be explored.

Other limitations include:

- **Data dependency**: Quality of the ORM is bounded by the quality of preference labels
- **Binary preferences**: Does not model more fine-grained preference structures (e.g., multiple preference levels, partial ordering)
- **Static evaluation**: Does not assess performance in active learning or online settings

9.2 Future Directions

Future work will investigate:

- **Multi-template robustness**: Evaluation across diverse reasoning formats and domains
- **Downstream integration**: Performance when used for best-of-N sampling, tree search, or iterative refinement in agentic systems
- **Integration with PRMs**: Combining outcome and process rewards for comprehensive reasoning evaluation in solver–reviewer–refiner agent loops
- **Cross-domain transfer**: Generalization to new task distributions
- **Scaling properties**: How performance changes with model size, training data, and training budget
- **Active learning**: Using ORM uncertainty to guide data collection
- **Multi-objective optimization**: Balancing correctness with other desiderata (efficiency, interpretability)

Of particular interest is integration with MagiCore-Agentic and similar frameworks, where outcome rewards can complement process-level supervision to guide multi-step reasoning.

10 Conclusion

We present a stable and reproducible approach to Outcome Reward Modeling based on pairwise preference learning. Through rigorous evaluation, we demonstrate that a simple pairwise ORM can serve as a reliable foundation for agentic reasoning systems under minimal supervision.

Our key findings are:

- Pairwise ORMs achieve 96.3% accuracy in relative preference discrimination with tight confidence intervals
- Training converges stably within 800 optimization steps ($\sim$10 minutes on a single GPU)
- The model exhibits strong anti-symmetry and uncertainty calibration
- Performance is robust across reasoning trace lengths

- The approach is practical, reproducible, and suitable for integration with existing agentic frameworks

These results suggest that pairwise preference learning is a practical and principled approach to reward modeling for agentic systems. The simplicity and stability of the method make it accessible to a wide range of applications, from mathematical reasoning to code generation to general problem-solving.

Data Availability

The trained model and dataset are publicly available on Hugging Face.[1]

Acknowledgments

I thank the broader research community working on reward modeling and agentic reasoning for their foundational contributions. Special appreciation to the authors of MagiCore-Agentic for their inspiring work on multi-step agentic systems.

References

[1] Liu, Z., Zhang, Y., Chen, H., et al. *MagiCore-Agentic: Robust Multi-Step Reasoning through Agentic Orchestration*. arXiv preprint arXiv:2409.12147, 2024. 1, 2

[2] Cobbe, K., Kosaraju, V., Bavarian, M., et al. *Training Verifiers to Solve Math Word Problems*. arXiv preprint arXiv:2110.14168, 2021. 1, 2

[3] Uesato, J., Kushman, N., Kumar, R., et al. *Solving Math Word Problems with Process- and Outcome-based Feedback*. arXiv preprint arXiv:2211.14275, 2022. 1

[4] Christiano, P., Leike, J., Brown, T., et al. *Deep Reinforcement Learning from Human Preferences*. Advances in Neural Information Processing Systems, 2017. 2

[5] Stiennon, N., Ouyang, L., Wu, J., et al. *Learning to Summarize from Human Feedback*. Advances in Neural Information Processing Systems, 2020. 2

[6] Ouyang, L., Wu, J., Jiang, X., et al. *Training Language Models to Follow Instructions with Human Feedback*. Advances in Neural Information Processing Systems, 2022. 2

[7] Le, H., Wang, Y., Gotmare, A. D., et al. *CodeRL: Mastering Code Generation through Pretrained Models and Deep Reinforcement Learning*. Advances in Neural Information Processing Systems, 2022. 2

[1] https://huggingface.co/LossFunctionLover/pairwise-orm-model
https://huggingface.co/datasets/LossFunctionLover/orm-pairwise-preference-pairs