

Every Question Has Its Own Value: Reinforcement Learning with Explicit Human Values

Dian Yu¹, Yulai Zhao^{1,2}, Kishan Panaganti¹, Linfeng Song¹, Haitao Mi¹, and Dong Yu¹

¹Tencent AI Lab

²Princeton University

Abstract

We propose Reinforcement Learning with Explicit Human Values (RLEV), a method that aligns Large Language Model (LLM) optimization directly with quantifiable human value signals. While Reinforcement Learning with Verifiable Rewards (RLVR) effectively trains models in objective domains using binary correctness rewards, it overlooks that not all tasks are equally significant. RLEV extends this framework by incorporating human-defined value signals directly into the reward function. Using exam-style data with explicit ground-truth value labels, RLEV consistently outperforms correctness-only baselines across multiple RL algorithms and model scales. Crucially, RLEV policies not only improve value-weighted accuracy but also learn a value-sensitive termination policy: concise for low-value prompts, thorough for high-value ones. We demonstrate this behavior stems from value-weighted gradient amplification on end-of-sequence tokens. Ablation studies confirm the gain is causally linked to value alignment. RLEV remains robust under noisy value signals, such as difficulty-based labels, demonstrating that optimizing for an explicit utility function offers a practical path to aligning LLMs with human priorities.

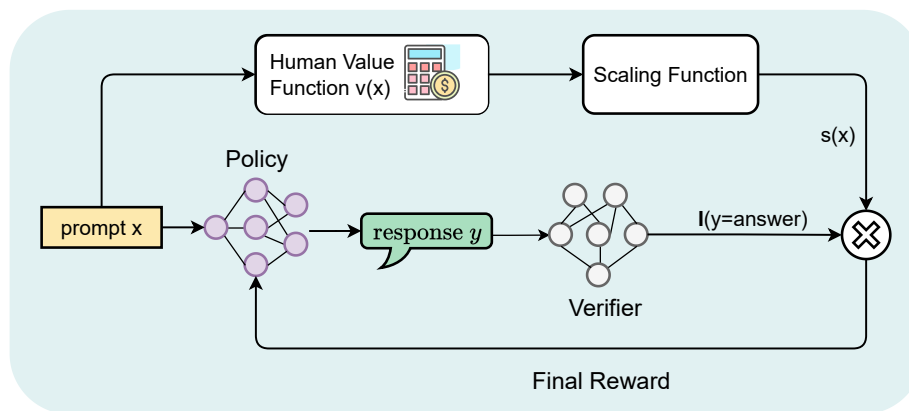


Figure 1: RLEV overview. The verifier can be either a reward model or rule-based function.

1 Introduction

Aligning Large Language Models (LLMs) with human goals can follow two paradigms: **implicit** value learning, which infers human utility from feedback, and **explicit** value learning, which optimizes directly for defined utility signals. The dominant paradigm, Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al. (2020); Ouyang et al. (2022); Rafailov et al. (2023)), learns an **implicit** utility model from subjective pairwise preferences. While effective for non-verifiable tasks, this is often unnecessary for objective domains. For these, Reinforcement Learning with

Verifiable Rewards (RLVR) (Lambert et al. (2024); Guo et al. (2025); Su et al. (2025)) offers a simpler, more direct approach, using a binary reward for correctness. However, this method carries a critical oversight: by assigning a uniform reward (e.g., +1) to all correct answers, it treats all prompts as equally important, failing to capture the explicit and non-uniform value common in real-world scenarios. For instance, on an exam, correctly answering a 10-point question is demonstrably more valuable than answering a 2-point one. An LLM trained to maximize only the count of correct answers is not optimized for the total score, which is the true human objective.

To bridge this gap, we introduce **Reinforcement Learning with Explicit Human Values (RLEV)**, a method that extends the RLVR framework by integrating explicit, human-assigned values into the reward function. RLEV operationalizes a simple principle: the utility of a response depends jointly on its correctness and the intrinsic value of its prompt. Using 100k exam-style training examples with ground-truth value labels, we show that RLEV consistently outperforms the standard RLVR baseline across multiple RL algorithms (REINFORCE++ (Hu, 2025), RLOO (Ahmadian et al., 2024), and GRPO (Shao et al., 2024)) and model scales (7B and 32B) (Team, 2024). Notably, RLEV-trained policies learn a value-sensitive termination policy, generating highly concise responses for low-value prompts while remaining thorough on high-value ones. Our gradient analysis reveals this behavior stems from the value-scaled reward amplifying updates on end-of-sequence tokens, encouraging the model to terminate efficiently based on the prompt’s importance.

Crucially, we demonstrate through ablation studies that this performance gain is causally linked to alignment with human-defined values. Baselines using randomly shuffled or uniformly scaled rewards show no significant improvement over correctness-only training. Finally, we show RLEV is robust even with noisy value signals, such as pseudo-labels from a score predictor or weak labels based on question difficulty, which still outperform the baseline. These findings establish that directly optimizing for an explicit utility function is a potent and effective method for aligning LLM behavior with stated human priorities.

Our contributions are as follows:

- We propose RLEV, a novel training paradigm that aligns LLMs with explicit human priorities by scaling correctness rewards with quantifiable value signals.
- We demonstrate empirically that RLEV consistently outperforms strong correctness-only baselines across multiple RL algorithms and model scales, leading to higher value-weighted scores and a desirable property of generating more concise responses.¹
- Through gradient analysis and ablation studies, we provide strong evidence that these gains are causally linked to value alignment, not merely to changes in reward magnitude.
- We show that RLEV is robust and practical, achieving superior performance even when using noisy or approximate value signals, such as difficulty-based weak labels.

2 Method: Learning from Human-Aligned Rewards

To align a Large Language Model with human priorities, we first define a utility function that captures the desired behavior. We then operationalize this function as a scalar reward for reinforcement learning.

2.1 A Human Utility Function for Valued and Verifiable Outcomes

We begin from the principle that the value of a model’s response depends on both its correctness and the importance of the prompt. We can formalize this by defining a **human utility function**, $U(x, y)$, for a response y to a prompt x :

$$U(x, y) = v(x) \cdot \mathbf{1}_{\text{correct}}(y) \quad (1)$$

where:

¹The RLEV dataset is available at <https://huggingface.co/datasets/sarosavo/RLEV>.

- $v(x)$ represents the intrinsic **human-defined value** or importance of the prompt x .
- $\mathbf{1}_{\text{correct}}(y)$ is an indicator function that is 1 if the response y is verifiably correct and 0 otherwise.

This utility function captures the simple, powerful idea that a correct response is worth the value of the question, while an incorrect response has zero utility. The goal of our alignment process is to train a policy π that maximizes the expected utility, $\mathbb{E}_{y \sim \pi(y|x)}[U(x, y)]$. While we instantiate this principle in exam-like settings, the same formulation applies to any domain where outcome correctness and human-assigned importance jointly determine utility, such as medical triage, tutoring, or content moderation. This product-based utility function $U(x, y)$ is a straightforward formalization of human priorities in domains where outcome correctness is verifiable and input importance is non-uniform (e.g., exams, medical triage).

2.2 Normalizing Human Values

To obtain a practical signal for $v(x)$, we use ground-truth scores from human-designed tasks, such as exams. Since different exams have different total scores, we must normalize these values to create a consistent scale. Let:

- s_{ij} be the raw score of question j in exam i .
- T_i be the total score of exam i .

We define the normalized value $v(x)$ for a given question x (i.e., question j in exam i) as its proportion of the exam total:

$$v(x) = \frac{s_{ij}}{T_i} \quad (2)$$

This proportional scaling naturally bounds $v(x)$ between 0 and 1 and makes it interpretable as the relative importance of the question.

2.3 The RLEV Reward Function

While $U(x, y)$ defines our target objective, its direct use as a reward can lead to unstable training. A low-value but correct question could receive a near-zero reward, discouraging the model from learning to answer it. To ensure a stable and effective learning signal, we design a practical surrogate reward function, $r(x, y)$, that preserves the relative importance of prompts while guaranteeing a minimum reward of 1 for any correct response. We achieve this by defining a scaling factor $s(x)$ based on the normalized human value $v(x)$ that is always greater than or equal to 1:

$$r(x, y) = s(x) \cdot \mathbf{1}_{\text{correct}}(y) \quad (3)$$

where $s(x)$ is a scaling factor based on the normalized human value $v(x)$:

$$s(x) = 1 + \min(\alpha \cdot v(x), 1) \quad (4)$$

Here, α is a scaling hyperparameter. The resulting reward $r(x, y)$ is within the range $[1, 2]$ for correct responses and is 0 for incorrect ones. This formulation incentivizes correctness on all questions while providing a stronger “bonus” for correctly answering high-value ones. This additive and clipped form is chosen specifically to ensure a stable learning signal by providing a minimum reward for all correct answers while preventing excessively large rewards from destabilizing the training process, a design choice validated in our ablation studies (Section 3.8).

2.4 The Reinforcement Learning Objective

We aim to find the optimal policy π_θ that maximizes the expected cumulative reward $J(\theta)$ over a dataset of prompts $\mathcal{D}$, standard in REINFORCE-style RL (Williams, 1992):

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[\sum_{t=0}^{T-1} r(x, y_{<t}, y_t) \right],$$

where $r(x, y_{<t}, y_t)$ denotes the per-step reward. In our setting, the reward is sparse and non-zero only at the final step T , thus simplifying the objective to:

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} [r(x, y)].$$

The corresponding gradient is:

$$\nabla J(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[r(x, y) \nabla \log \pi_\theta(y|x) \right].$$

Given that the policy is autoregressive, $\log \pi_\theta(y|x) = \sum_{t=0}^{T-1} \log \pi_\theta(y_t|x, y_{<t})$.

2.5 Gradient Derivation

To analyze the learning dynamics, we derive the policy gradients for a single prompt x (noting the full gradient $\nabla J(\theta)$ is the expectation over $\mathcal{D}$) with respect to the parameters at a single time step t . At step t , we use z_k to refer to the logit at token $k \in \mathcal{V}$ where $\mathcal{V}$ denotes the whole vocabulary. Note that $\mathcal{V}$ also includes EOS symbol, denoted as e . For any token $v \in \mathcal{V}$, we use the following to represent the conditional probability that *the final y is correct* given $y_t = v$, where the probability is taken over the remaining rollout under the current policy

$$p_v = \Pr(\text{correct} \mid x, y_{<t}, y_t = v). \quad (5)$$

Logits at step t are converted to probabilities using the softmax function; then we have

$$\begin{aligned} \frac{\partial}{\partial z_k} \log \pi(y_t \mid x, y_{<t}) &= \frac{\partial}{\partial z_k} \log \frac{\exp(z_{y_t})}{\sum_{v \in \mathcal{V}} \exp(z_v)} \\ &= \mathbf{1}\{y_t = k\} - \pi(k \mid x, y_{<t}). \end{aligned} \quad (6)$$

$$\begin{aligned} \frac{\partial J}{\partial z_k} &= \mathbb{E}_{y_t \sim \pi(\cdot|x, y_{<t})} \left[r(x, y) (\mathbf{1}\{y_t = k\} - \pi(k \mid x, y_{<t})) \right] \\ &= \mathbb{E}[r(x, y) \mathbf{1}\{y_t = k\}] - \pi(k \mid x, y_{<t}) \mathbb{E}[r(x, y)] \\ &= \pi(k \mid x, y_{<t}) (\mathbb{E}[r(x, y) \mid y_t = k] - \mathbb{E}[r(x, y)]) \end{aligned} \quad (7)$$

where we employ the law of total expectation in the last line.

Since $r(x, y) = s(x) \cdot \mathbf{1}_{\text{correct}}(y)$, as $s(x)$ is constant for a given x , we have

$$\mathbb{E}[r(x, y) \mid y_t = k] = s(x) p_k \quad (8)$$

$$\mathbb{E}[r(x, y)] = s(x) \sum_{v \in \mathcal{V}} \pi(v \mid x, y_{<t}) p_v \quad (9)$$

Therefore,

$$\begin{aligned}
\frac{\partial J}{\partial z_k} &= \pi(k \mid x, y_{<t}) (\mathbb{E}[r \mid y_t = k] - \mathbb{E}[r]) \\
&= \pi(k \mid x, y_{<t}) (s(x) p_k - s(x) \sum_{v \in \mathcal{V}} \pi(v \mid x, y_{<t}) p_v) \\
&= \boxed{\pi(k \mid x, y_{<t}) s(x) \cdot (p_k - \sum_{v \in \mathcal{V}} \pi(v \mid x, y_{<t}) p_v)}
\end{aligned} \tag{10}$$

Consider the special EOS token e , which is also in $\mathcal{V}$. Below, we investigate the training dynamics of this special token. First we define $\bar{p}_{-e}$ as the averaged probability of final correctness over all non-EOS tokens:

$$\bar{p}_{-e} := \frac{1}{1 - \pi_e} \sum_{v \neq e, v \in \mathcal{V}} \pi_v p_v. \tag{11}$$

Then the advantage for choosing EOS is

$$\mathbb{E}[r \mid y_t = e] - \mathbb{E}[r] = s(x)(1 - \pi_e)(p_e - \bar{p}_{-e}). \tag{12}$$

Thus the gradient with respect to the EOS logit is:

$$\boxed{\frac{\partial J}{\partial z_e} = s(x) \cdot \pi_e(1 - \pi_e)(p_e - \bar{p}_{-e})}. \tag{13}$$

The resulting gradient for the EOS logit (Equation 13) is driven by the difference between the expected correctness of terminating the sequence (p_e) and the average expected correctness of continuing ($\bar{p}_{-e}$), scaled by the human value factor s .

- EOS receives a positive gradient if its correctness probability exceeds the average continuation correctness, i.e. $p_e > \bar{p}_{-e}$.
- A continuation token c receives a negative gradient if $p_c < \sum_v \pi_v p_v$ (the policy-weighted average). Thus, when $p_e > \bar{p}_{-e}$, most continuations are pushed down, though any c with p_c above the global average can still receive a positive update.
- The human-aligned scale $s \in [1, 2]$ multiplies the gradient magnitude without changing these conditions. Therefore, when $p_e > \bar{p}_{-e}$ (i.e. there is already sufficient information for correctness), EOS is reinforced more strongly, which accelerates the tendency to end earlier.

In summary, compared to a purely binary correctness reward, this scheme encourages the policy to generate more concise, more accurate completions. Moreover, because the reward is scaled by the human-defined scoring function rather than correctness alone, the resulting policy is expected to achieve higher human-defined scores in real-world use, which is supported by our experimental results in Section 3. **Multiplying by the human-value factor amplifies the gradient’s magnitude, which more strongly reinforces the decision to terminate when correctness is already likely.**

3 Experiments

3.1 Datasets

The dataset comprises question-answering pairs from multi-subject exams, with the original content predominantly in Chinese. The reference answers are written by domain experts for objective human evaluation, making them suitable for RLVR. Additionally, we extract each question’s human-labeled score and the total score of the exam it originates from. Subsequently, we partition the data into

training and testing sets containing 100,000 and 8,000 examples, respectively. We split by exam to avoid leakage.

To assess the generalization ability of the RLEV policies trained on Chinese data, we evaluate the out-of-distribution performance on several English and Chinese general-domain benchmarks (GPQA Diamond (Rein et al., 2024), C-Eval (Huang et al., 2023), MMLU-Pro (Wang et al., 2024), and SuperGPQA (Du et al., 2025)).

As ground-truth human-defined values may be unavailable in many scenarios, in Section 3.5, we investigate the effectiveness of RLEV with two types of “noisy” human values. We conduct experiments using WebInstruct-verified (Ma et al., 2025), a general domain English dataset with objective answers. We map each of the five difficulty category (PRIMARY SCHOOL, JUNIOR HIGH SCHOOL, SENIOR HIGH SCHOOL, UNIVERSITY, and PHD) into value scores (1, 2, 4, 6, 8), respectively. We divide the score by 100 for normalization. For each category, we randomly sample 2,000 training examples and train with the resulting 10k instances.

To make this resource more accessible to the broader research community, we used GPT-4o to translate the data (with human-labeled values) into English, which will also be released.

3.2 Experimental Setup

We kept the training setup consistent for all estimators. All policies were trained for one epoch on eight GPUs with a learning rate of $5e-7$. The rollout batch size was set to 128. The maximum length for both prompts and generated responses was capped at 1024 tokens. For evaluation, we use greedy decoding. We use base models (Qwen2.5-7B and Qwen2.5-32B (Team, 2024)) for policy initialization.

3.3 Evaluation Metrics

To evaluate our method, we use a set of metrics designed to capture both correctness and alignment with human-defined values:

Accuracy (Acc): The standard, unweighted accuracy calculated as the percentage of total correct responses. This metric measures correctness without considering the value of each prompt.

Human-Aligned Accuracy (H-Acc): The value-weighted accuracy, calculated as the ratio of achieved value from correct responses to the total possible value:

$$\text{H-Acc} = \frac{\sum_{\text{correct responses}} v(x)}{\sum_{\text{all responses}} v(x)}$$

Response Length (Resp. Length): The average number of tokens in a model’s generated response.

Value Density: An efficiency metric measuring value delivered per token, calculated by dividing the **H-Acc value expressed as a percentage** by the average response length. This is particularly relevant for tasks focused on verifiable correctness, where the primary goal is to provide the correct answer efficiently.

Following previous RLVR studies for general domains (Su et al., 2025; Ma et al., 2025), we use a large language model (Qwen2.5-72B-Instruct (Team, 2024)) to verify the semantic equivalence between the final answer of a response and the reference answer. This automated verification method has been widely shown to have high agreement with human annotators in objective, non-adversarial, reference-based evaluation settings (Zhao et al., 2025). Importantly, focusing verification on only the final part of the response did not cause length collapse in our experiments.

3.4 RLEV with Ground-Truth Human Values

Our primary results show that RLEV consistently outperforms the correctness-only baseline across all tested configurations. This holds true for both 7B and 32B models, which see average Human-Aligned Accuracy (H-Acc) gains of 2.0% and 2.8%, respectively (Table 1). This improvement is driven by a learned focus on high-value tasks; as detailed in the appendix (Table 8), the accuracy gains are generally notably larger for high-valued prompts than for low-valued ones.

A key benefit is a value-sensitive termination policy, which will be discussed in Section 3.6. The model learns to be concise on low-value prompts while remaining thorough on high-value ones. This leads to an overall increase in conciseness. For example, RLEV models more than halve the average response length, from 246.9 to 98.6 tokens for the 32B models.

This efficiency and strategic improvement also generalize effectively. Even though trained on Chinese data, the RLEV models outperform their correctness-only counterparts on several out-of-distribution (OOD) English and Chinese benchmarks, with the 32B model showing notable gains on tasks like GPQA Diamond and SuperGPQA (Table 2).

Table 1: Comparison of policies trained with RLEV (human-aligned) and baseline (correctness) rewards across 7B and 32B models. RLEV consistently improves accuracy and conciseness for both model scales.

Estimator	Size	Reward Type	Acc	H-Acc	Resp. Length	Value Density
REINFORCE++	7B	correctness	63.8	55.0	168.1	0.33
		human-aligned	65.3	57.0	84.8	0.67
	32B	correctness	67.7	57.6	226.2	0.25
		human-aligned	71.0	61.9	68.7	0.90
RLOO	7B	correctness	65.9	56.7	186.2	0.30
		human-aligned	66.6	58.9	86.4	0.68
	32B	correctness	70.9	60.9	345.5	0.18
		human-aligned	72.3	63.3	78.7	0.80
GRPO	7B	correctness	65.7	56.0	251.1	0.22
		human-aligned	66.2	57.7	100.4	0.57
	32B	correctness	70.6	59.9	169.0	0.35
		human-aligned	71.3	61.7	148.3	0.42
Average	7B	correctness	65.1	55.9	201.8	0.28
		human-aligned	66.0	57.9	90.5	0.64
	32B	correctness	69.7	59.5	246.9	0.26
		human-aligned	71.5	62.3	98.6	0.71

Table 2: OOD Results across English and Chinese general-domain tasks.

Model	GPQA Diamond	C-Eval	MMLU-Pro	SuperGPQA
Base-7B	31.8	60.8	45.0	25.4
+ correctness	31.8	76.2	51.5	26.2
+ human-aligned	31.3	76.4	52.5	26.8
Base-32B	33.2	57.9	55.1	33.2
+ correctness	39.9	84.9	63.0	34.0
+ human-aligned	43.4	85.4	63.0	36.2

3.5 RLEV with Other Types of Human Values

Table 3: RLEV performance with imperfect value signals on the test set of WebInstruct-verified. We test two “noisy” value sources: weak labels derived from task difficulty and predictor-generated values from a score predictor trained on our main exam dataset. Both methods consistently outperform the correctness-only baseline, showing RLEV’s robustness when ground-truth values are unavailable.

Model	Acc H-Acc	primary	junior	senior	university	phd
Base-7B	18.8 17.0	38.9	28.4	24.5	13.9	0.0
REINFORCE++						
+ correctness	19.1 16.9	50.0	27.0	27.3	12.7	0.0
+ weak-labeled values	21.2 19.3	38.9	28.4	29.4	15.3	0.0
+ predicted values	21.6 19.6	50.0	32.4	27.9	15.8	10.0
RLOO						
+ correctness	20.0 18.0	38.9	28.4	28.2	13.9	0.0
+ weak-labeled values	20.3 18.4	33.3	29.7	28.5	14.1	10.0
+ predicted values	21.3 19.1	38.9	32.4	30.0	14.6	0.0
GRPO						
+ correctness	19.4 17.0	44.4	33.8	27.0	12.8	0.0
+ weak-labeled values	20.6 18.7	50.0	31.1	25.5	15.8	0.0
+ predicted values	20.3 18.2	50.0	25.7	28.8	14.1	0.0

This result (Table 3) demonstrates RLEV’s robustness and practicality. It shows the method is effective even when precise, ground-truth scores are unavailable, making it applicable to a much wider range of real-world scenarios where only heuristic value estimates (like task priority or difficulty) exist. Note that primary and phd only have 18 and 10 instances, respectively, while the total test set has 1,000 instances. We use the multi-subject exam data for training a score predictor for generating the predicted values in Table 3. We discuss the training details in Appendix A.2.

3.6 Analysis of Value-Sensitive Termination

Token-level Analysis: The hypothesis from the gradient analysis (“value-scaling amplifies updates on the EOS token”) is directly validated by our token-level analysis, though the behavior is more nuanced than a simple uniform increase in EOS probability. As shown in Figure 2, the RLEV model learns a sophisticated, value-sensitive termination policy

For **low-value** prompts, the RLEV model assigns a dramatically higher probability to the EOS token much earlier in the generation process compared to the baseline. Once a sufficient answer is generated for these simpler prompts, the value-weighted reward strongly reinforces the decision to stop, leading to highly concise outputs.

Conversely, for **high-value** prompts, which are often more complex, the RLEV model learns to suppress the probability of the EOS token relative to the baseline. This behavior encourages the model to generate a more thorough and complete response. The gradient analysis explains this as the large value-scaling factor s amplifying the signal to continue when the expected correctness from adding more tokens is higher than from stopping prematurely.

This dual mechanism shows that RLEV does not merely learn to be shorter; it learns to allocate its token budget strategically, being efficient on low-stakes questions while being cautious and comprehensive on high-stakes ones to maximize the overall human-aligned score.

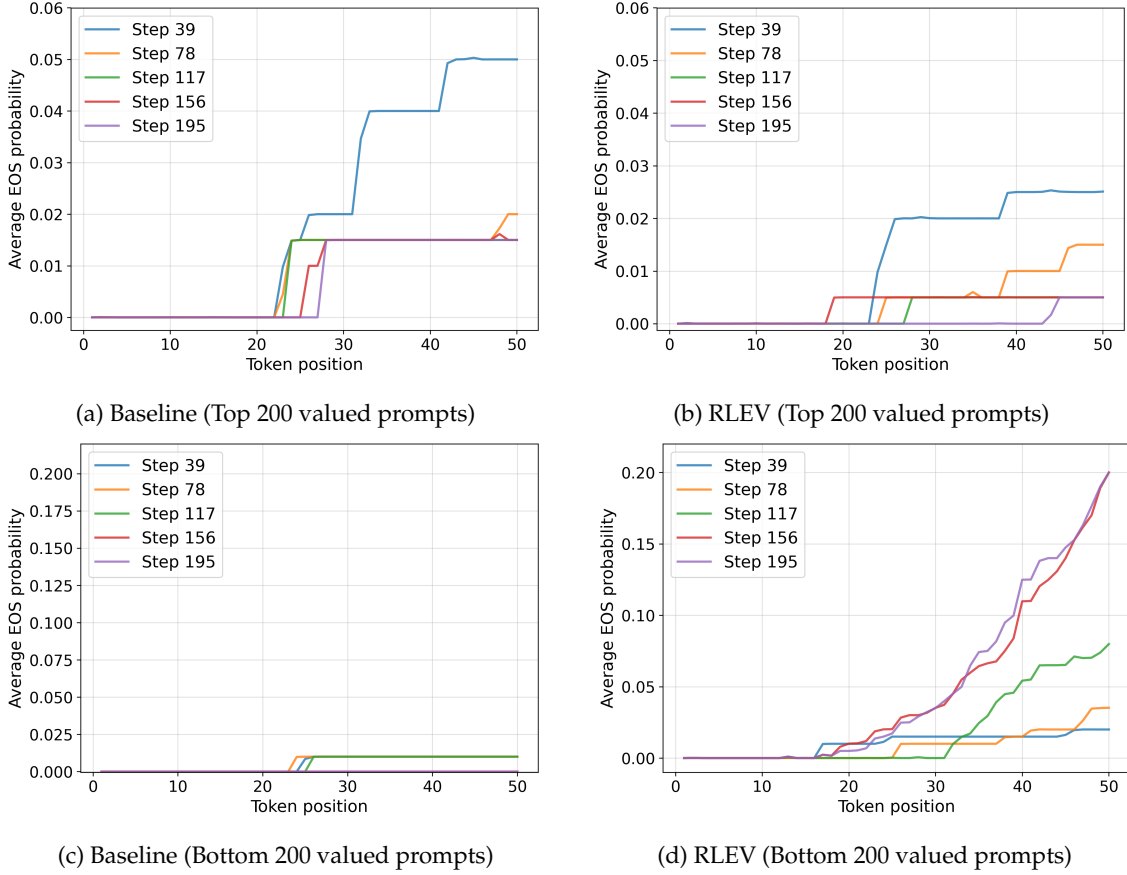


Figure 2: EOS probability trajectories for RLEV and the baseline, showing different termination policies for high-value (top) and low-value (bottom) prompts.

3.7 Ablation Studies: Isolating the Impact of Human Values

A key claim of our work is that aligning the reward signal with human-defined values is responsible for the observed performance gains. However, an alternative hypothesis is that the improvements stem from simply increasing the magnitude of the rewards for correct answers, rather than the value-alignment itself. To isolate the effect of human-aligned values, we conduct several ablation studies. Besides the correctness-only baseline, we also compare our full RLEV (human-aligned) model against two controls:

Uniform Scaling: All correct responses receive a constant $\bar{s}$, where $\bar{s}$ is the average reward scale calculated across the training prompts ($\bar{s} = \mathbb{E}[s(x)] \approx 1.2$) (details in Appendix A.1). This control is designed to test the alternative hypothesis that **general increase in reward magnitude, irrespective of its alignment with prompt value, is sufficient to cause the observed performance gains.**

Random Weights: The reward is scaled using the RLEV formula, but the human values $v(x)$ are randomly shuffled across the training set before being used to calculate the scaling factor $s(x)$. This procedure creates a placebo reward signal that maintains the exact same distribution of reward magnitudes as the primary experiment but completely decouples the reward from the prompt’s true value. This directly tests whether the **causal factor for improvement is the specific alignment** between higher rewards and higher-value prompts.

Table 4: Ablation study results using the RLOO estimator. Uniformly scaling the reward degrades performance, while using random weights does not improve conciseness. Only when the reward scaling is directly correlated with human-defined values do we see a meaningful increase in human-aligned accuracy (h-acc) and a desirable reduction in response length.

Reward Scaling Method	Acc	H-Acc	Resp. Length	Value Density
correctness (baseline)	65.9	56.7	186.2	0.30
uniform scaling	65.3	55.1	358.4	0.15
random weights (shuffled)	66.4	57.4	280.5	0.20
human-aligned (ours)	66.6	58.9	86.4	0.68

3.8 Reward Function Sensitivity and Design

To validate our reward function design, we analyze its sensitivity to both the hyperparameter α and its specific mathematical form.

Sensitivity to Hyperparameter α The choice of α is crucial as it determines how strongly the human value $v(x)$ influences the final reward. We trained models using our primary reward function, $r(x, y) = (1 + \min(\alpha \cdot v(x), 1)) \cdot \mathbf{1}_{\text{correct}}(y)$, with several values of α . As shown in Table 5, while performance is robust across a range of values, we found that $\alpha = 10$ offered the best balance of human-aligned accuracy and response conciseness.

Table 5: Sensitivity to hyperparameter α . Performance is reported across all key metrics.

Hyperparameter α	Acc	H-Acc	Resp. Length	Value Density
baseline	65.9	56.7	186.2	0.30
1	66.4	58.1	101.5	0.57
5	66.1	56.8	141.0	0.40
10	66.6	58.9	86.4	0.68
15	66.3	58.1	62.4	0.93
20	66.1	56.8	157.9	0.36

Ablation on Reward Function Form To justify our choice of an additive and clipped reward scaler, we compare it against a purely multiplicative alternative: $r(x, y) = (1 + \alpha \cdot v(x)) \cdot \mathbf{1}_{\text{correct}}(y)$. Table 6 shows that our chosen form yields superior results.

There are two possible reasons: first, the mean $v(x)$ is 0.02, and only 1.18% of the training examples have a value > 0.1 . This highly right-skewed distribution, which is visualized in Appendix A.1 (Figure 3), indicates that for over 98% of the data, our function acts as a fine-grained linear reward scaler, preserving the original human value. Second, for the small fraction of high-value outliers shown in the distribution’s tail, the clipping mechanism prevents the excessively large rewards that the purely multiplicative form would generate, thus stabilizing training process and leading to better overall performance.

Table 6: Comparison of different reward scaling functions.

Reward Function Form	Acc	H-Acc	Resp. Length	Value Density
purely multiplicative	66.4	57.6	201.6	0.29
additive & clipped (ours)	66.6	58.9	86.4	0.68

4 Related Work

The idea of weighting learning signals according to their relative importance has deep roots in classical RL. Early methods such as importance-weighted transfer (Tirinzoni et al., 2018), reward-weighted regression (Peters & Schaal, 2007), and advantage-weighted regression (Peng et al., 2019) all adjust gradient updates to emphasize high-value samples. These studies show that non-uniform weighting can improve sample efficiency or align behavior with desired utility, but they do not consider the case where each data point (e.g., a prompt or question) carries a human-defined point value reflecting its real-world importance.

Recent work in the LLM alignment domain has focused on RL with Verifiable Rewards (Luong et al., 2024; Lambert et al., 2024; Guo et al., 2025; Su et al., 2025), which train models using objective correctness signals. Other studies have proposed shaping or enriching verifiable rewards: for example, ConfClip (Zhang et al., 2025), rubrics as rewards (Gunjal et al., 2025), and composite reward frameworks such as RLCR with calibration rewards (Damani et al., 2025). While these approaches modify reward form or composition, they do not explicitly scale correctness rewards by human-assigned per-prompt values normalized across a dataset, nor analyze the resulting gradient-level mechanisms.

Our method, RLEV, integrates human-assigned per-prompt importance into the RLVR framework using a clipped scaling surrogate. Through empirical tests, ablations, and gradient analysis, RLEV yields more concise and human-aligned behavior by optimizing for explicit, value-weighted utility, enabling alignment with explicitly defined human utility functions.

5 Conclusions and Future Work

We introduced Reinforcement Learning with Explicit Human Values (RLEV), a paradigm that aligns LLMs with human priorities by scaling correctness rewards with an explicit value signal. Experiments show RLEV consistently outperforms correctness-only baselines, improving value-weighted accuracy and leading to the generation of more concise responses. We trace this conciseness to value-weighted gradient amplification on end-of-sequence (EOS) tokens. Ablation studies confirm these gains are causally linked to value alignment rather than reward magnitude. Furthermore, the method proves robust, surpassing baselines even with noisy value signals derived from task difficulty.

Future work could explore more dynamic value functions that are learned or adapt to user priorities. Another promising direction is to combine RLEV, for grounding in objective correctness and importance, with RLHF to fine-tune subjective qualities like style and tone. This hybrid approach could offer a more holistic path to LLM alignment.

6 Broader Impact and Limitations

Ultimately, this work demonstrates that directly optimizing for an explicit, non-uniform utility function is a robust and effective method for aligning LLM behavior with human priorities. By moving beyond simple binary rewards, RLEV encourages models to develop a more nuanced understanding of value, learning not just what constitutes a correct answer, but also how much each correct answer matters. This represents a practical step toward creating systems that are not only more capable but also more judicious in applying their capabilities to what humans deem most important. Despite its effectiveness, this work has several limitations. First, the framework formulates human value as a single, pre-defined scalar quantity suited for objective domains where importance is explicitly quantified. However, human values in a broader sense are often complex, multi-dimensional, and subjective. Second, applying RLEV requires explicit value labels for each prompt. While our experiments show RLEV is robust to noisy signals, this data dependency remains a practical consideration. Finally, the current method relies on a static value function, and future work could explore more dynamic value functions that adapt to user priorities in real-time.

References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.
- Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*, 2025.
- Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.
- Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.
- Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36: 62991–63010, 2023.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.
- Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 2024.
- Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhui Chen. General-Reasoner: Advancing llm reasoning across all domains. *arXiv:2505.14652*, 2025. URL <https://arxiv.org/abs/2505.14652>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.
- 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. *Advances in neural information processing systems*, 36:53728–53741, 2023.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.
- Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025.
- Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL <https://qwenlm.github.io/blog/qwen2.5/>.
- Andrea Tirinzoni, Andrea Sessa, Matteo Pirota, and Marcello Restelli. Importance weighted transfer of samples in reinforcement learning. In *International Conference on Machine Learning*, pp. 4936–4945. PMLR, 2018.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.
- Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.
- Bonan Zhang, Zhongqi Chen, Bowen Song, Qinya Li, Fan Wu, and Guihai Chen. Confclip: Confidence-weighted and clipped reward for reinforcement learning in llms. *arXiv preprint arXiv:2509.17730*, 2025.
- Yulai Zhao, Haolin Liu, Dian Yu, Sunyuan Kung, Meijia Chen, Haitao Mi, and Dong Yu. One token to fool llm-as-a-judge. *arXiv preprint arXiv:2507.08794*, 2025.

A Appendix

A.1 Data Statistics

We analyze the human-defined values in 100k training instances and the 8k testing instances. These scores are normalized per-exam proportional scores defined in Equation 2 (Section 2.2). See distribution of normalized values in the training and test subsets in Figure 3 and Figure 4.

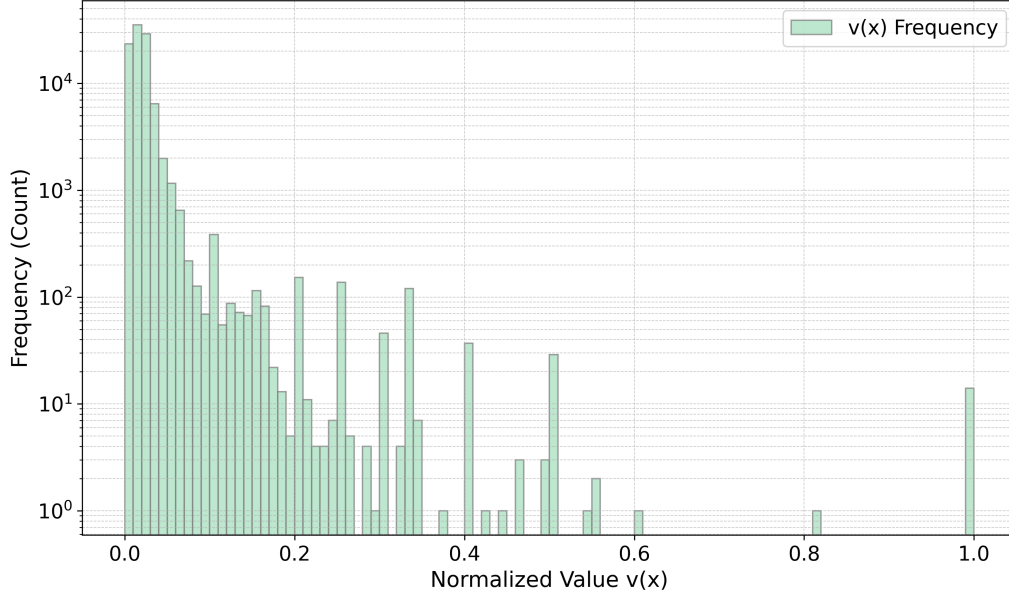


Figure 3: Distribution of human-defined normalized values $v(x)$ in training data (100k) with ground-truth values.

A.2 Score Prediction

Table 7: Prompt Structure for the Score Predictor

Role	Content
system	f“You are a scoring assistant. Given a question and its answer, output a numeric score greater than 0 and up to {total_score} inclusive (decimals allowed, e.g. 0.5) that reflects how much this problem would contribute in a {total_score}-point exam. Respond with only the score, no other text.”
user	question

To evaluate RLEV’s performance with imperfect value signals, besides rule-based scores derived from difficulty levels, we train a score predictor to generate pseudo values for datasets where ground-truth scores are unavailable.

We convert the exam data into the format shown in Table 7 and train the score predictor with supervised fine-tuning for two epochs using Qwen2.5-7B. For datasets such as WebInstruct-verified, we standardize the task by setting a consistent total score of 100 for all prompts. We use the same test set for evaluating the performance of the score predictor. It achieves an exact-match accuracy of

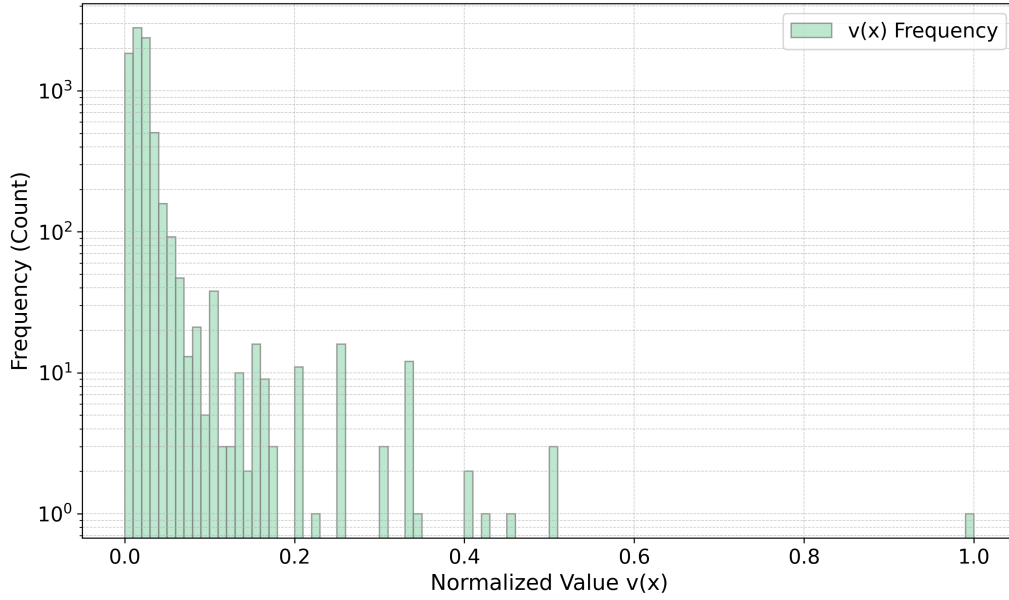


Figure 4: Distribution of human-defined normalized values $v(x)$ in test data (8k) with ground-truth values.

79.5%. The Pearson correlation between the predicted and true scores is 0.91 ($p < 0.001$), indicating a strong positive relationship.

A.3 Detailed Accuracy Analysis

Table 8: Comparison of policies trained with RLEV (human-aligned) and baseline (correctness) rewards across 7B and 32B models. We also report the accuracy on top 20% high-valued prompts and bottom 20% low-valued prompts.

Estimator	Size	Reward Type	Acc (all)	Acc (high-valued)	Acc (low-valued)
REINFORCE++	7B	correctness	63.8	54.5	68.9
		human-aligned	65.3	58.0	69.8
	32B	correctness	67.7	57.6	73.4
		human-aligned	71.0	62.9	76.2
RLOO	7B	correctness	65.9	57.4	71.6
		human-aligned	66.6	58.8	71.8
	32B	correctness	70.9	60.9	76.6
		human-aligned	72.3	62.1	78.1
GRPO	7B	correctness	65.7	55.7	71.9
		human-aligned	66.2	57.1	72.4
	32B	correctness	70.6	59.3	76.8
		human-aligned	71.3	61.0	76.6

As shown in Table 8, human-aligned (RLEV) policy achieves a higher accuracy than the correctness baseline in all high-valued bins and nearly all low-valued bins. The improvement is generally more obvious for the high-valued prompts. These results show that RLEV specifically guides the model to perform better on the prompts that are defined as more valuable or important.