LMUnit: Fine-grained Evaluation with Natural Language Unit Tests

Anonymous ACL submission

Abstract

As language models become integral to critical workflows, assessing their behavior remains a fundamental challenge – human evaluation is costly and noisy, while automated metrics provide only coarse, difficult-to-interpret signals. We introduce **natural language unit tests**, a paradigm that decomposes response quality into explicit, testable criteria, along with a unified scoring model, **LMUnit**, which combines multi-objective training across preferences, direct ratings, and natural language rationales. Through controlled human studies, we show this paradigm significantly improves inter-annotator agreement and enables more effective LLM development workflows. LMUnit achieves state-of-the-art performance on evaluation benchmarks (FLASK, BigGenBench) and competitive results on RewardBench. These results validate both our proposed paradigm and scoring model, suggesting a promising path forward for language model evaluation and development.

1 Introduction

The evaluation of generative language models remains one of the most fundamental challenges in natural language processing (Jones and Galliers, 1995; Deriu et al., 2021; Smith et al., 2022; Chang et al., 2024) – it determines how we measure progress and shapes the field's trajectory. As these models transition from research prototypes to production systems, users increasingly rely on them for critical workflows (Lin et al., 2024a), creating an urgent need for evaluation methods that identify response strengths/weaknesses, ensure reliability, and prevent costly regressions. Yet current approaches fall short: human evaluation is expensive and struggles to discern subtle differences among top models (Hosking et al., 2023; Clark et al., 2021; Karpinska et al., 2021), while automated metrics compress response quality into coarse scores (Stent et al., 2005; Liu et al., 2016) that rely on implicitly learned, often biased criteria

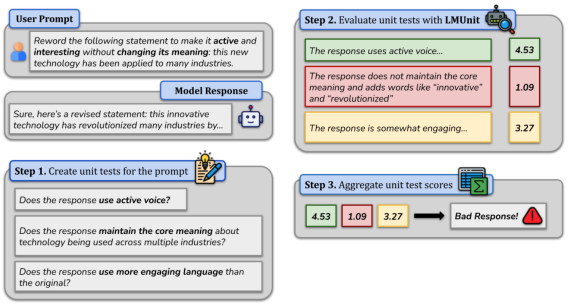

Figure 1: **Natural Language Unit Tests:** Overview of the three-step process: (1) unit test creation, (2) LMUnit-based scoring with natural language rationales, and (3) score aggregation for overall quality assessment.

(Dubois et al., 2024a; Shankar et al., 2024; Zhang et al., 2024a). As models become more deeply integrated into essential workflows, it is imperative that our evaluation methodologies evolve in tandem, empowering LLM practitioners to reliably **detect subtle failures**, meaningfully **distinguish among top-performing systems**, and **generate actionable insights** that drive improvements.

We focus on measuring response quality - one of the most critical challenges in evaluating language models. Defining "response quality" is inherently complex, depending on multiple factors including factual accuracy, logical coherence, and alignment with user objectives, which vary by domain, application, style, and context (Mehri and Eskenazi, 2020a; Ye et al., 2023; Krishna et al., 2023). Existing approaches struggle with this complexity: (1) reference-based comparisons fail in open-ended scenarios where no single "correct" response exists (Liu et al., 2016; Lowe et al., 2017), (2) human evaluations become inconsistent as models grow more capable and errors subtler (Walker et al., 2007; Pan et al., 2024; Christiano et al., 2023), and (3) preference models and LLM judges compress nuanced assessments into opaque metrics that are difficult to interpret or steer (Dubois et al., 2024b; D'Oosterlinck et al., 2024; Singhal et al., 2023).

To address these limitations, we propose **natural language unit tests**, a paradigm that decomposes response quality into explicit, testable criteria that humans can define, refine, and guide over time (Figure 1). While this approach enhances transparency, reliably scoring and integrating these fine-grained assessments while maintaining human values alignment remains a key challenge.

Building an effective scoring model for unit tests presents a significant challenge: it must accurately evaluate a wide range of criteria – ranging from broad notions of quality to detailed rubrics that capture intricate, context-specific requirements. Existing approaches each address part of the problem: prompted LLM judges can be instructed to consider certain criteria (Liu et al., 2023), but their accuracy is limited by generic instruction-following abilities and the inability to learn directly from preference data (Wang et al., 2024b; Zhong et al., 2022); preference models, while closely aligned with human judgments, lack promptability and struggle to handle more granular, human-defined criteria (Singhal et al., 2023; Lambert and Calandra, 2023).

To address these challenges, we propose LMUNIT, a unified modeling approach that optimizes large language models as preference models while supporting flexible, user-defined evaluation criteria. By combining diverse training signals with natural language rationales, LMUNIT achieves strong results across preference modeling, direct scoring, and fine-grained unit test evaluations, laying a robust foundation for more adaptive and transparent evaluation methodologies. These rationales are optional at inference time but enabling them allows further interpretability of scores.

To demonstrate our paradigm's effectiveness in enabling human stakeholder intervention, we assess its real-world impact through human studies. In a controlled annotation study, expert raters achieved higher inter-annotator agreement when evaluating outputs against explicit unit tests compared to standard preference annotations. Additionally, in a case study with LLM developers, LMUNIT's transparent, test-driven evaluations enabled identification of more errors than conventional LLM judges, demonstrating the value of our proposed paradigm

Our key contributions include: (1) proposing the paradigm of natural language unit tests, and validating it at scale, (2) developing LMUNIT as a unified scoring model that achieves state-of-the-art performance, (3) showing the benefits and challenges of effective unit test creation and weighting strategies, (4) demostrating the importance of rationales when incorporating them as part of the training data. (5) validating our approach through human studies that demonstrate improved inter-annotator agreement and more effective LLM development workflows.

2 Related Work

2.1 Evaluation of Generative Language Models

While human evaluation remains the gold standard for LLMs (Ouyang et al., 2022; Touvron et al., 2023), its scalability limitations (Hosking et al., 2023; Schoch et al., 2020) have driven the development of automated approaches. These include word overlap metrics (Papineni et al., 2002; Lin, 2004), embedding-based scoring (Yuan et al., 2021; Zhang et al., 2019), model-based evaluations (Lowe et al., 2017; Mehri and Eskenazi, 2020b; Zhong et al., 2022; Saad-Falcon et al., 2023), reward modeling (Christiano et al., 2017; Askell et al., 2021; Kim et al., 2023), and LM judges (Zheng et al., 2023; Liu et al., 2023; Es et al., 2023; Ravi et al., 2024; Kim et al., 2024a; Li et al., 2024b). However, automated methods often lack interpretability and can show biases that diverge from human judgments (Shankar et al., 2024; Wang et al., 2023b; Chaudhari et al., 2024). Recent work has focused on developing fine-grained evaluators (Ye et al., 2023; Wang et al., 2024b; Ribeiro et al., 2020; Lin and Chen, 2023; Cook et al., 2024) and unifying evaluation paradigms (Wang et al., 2024b; Kim et al., 2024c; Wu et al., 2023). For code generation specifically, LLM-based unit test generation has improved performance evaluation through compiler-compatible synthetic tests (Chen et al., 2022; Yuan et al., 2023; Saad-Falcon et al., 2024).

2.2 LM Judges

LLMs can be prompted to evaluate responses without additional training, showing high correlation with human ratings (Liu et al., 2023; Wang et al., 2023a; Fu et al., 2023; Chiang and Lee, 2023; Es et al., 2023; Kocmi and Federmann, 2023; Li et al., 2024a). While some approaches focus on in-context examples and evaluation instructions (Fu et al., 2023), others leverage chain-of-thought prompting (Liu et al., 2023) or fine-tune specialized judges (Saad-Falcon et al., 2023; Tang et al., 2024). However, these approaches face key limitations: poor generalization across evaluation tasks (Es et al., 2023; Saad-Falcon et al., 2023; Ravi et al., 2024) and systematic biases in position, verbosity,

and self-preference (Chen et al., 2024; Pan et al., 2024; Zheng et al., 2023; Koo et al., 2023).

2.3 Reward Models

Reward models, while widely adopted for evaluating and aligning language models (Bradley and Terry, 1952; Christiano et al., 2017; Liu and Zeng, 2024), face fundamental challenges: low inter-annotator agreement (65% - 75% - early RLHF papers) in human preference data (Askell et al., 2021; Ouyang et al., 2022; Wang et al., 2024a), noisy and inconsistent preferences (Dubois et al., 2024b), and spurious correlations like favoring longer responses (Lambert and Calandra, 2023; Singhal et al., 2023; Dubois et al., 2024a). Recent work shows promise in addressing these: Helpsteer-2 (Wang et al., 2023c) improved performance through better preference data collection. GenRM-COT (Zhang et al., 2024b) and EvalPlanner (Saha et al., 2025) used chain-of-thought reasoning for more reliable evaluation. However, challenges with reward underspecification and alignment persist (Eisenstein et al., 2023; Chaudhari et al., 2024).

2.4 Fine-Grained Evaluators

Breaking down complex evaluation problems has been foundational in NLP (Walker et al., 2000) and remains vital for language models (Saha et al., 2024). While early approaches used fixed evaluation dimensions (Liu et al., 2016; Lowe et al., 2017; Zhong et al., 2022), modern language models enable more dynamic, fine-grained criteria (Mehri and Eskenazi, 2020a; Lin and Chen, 2023; Ye et al., 2023; Kim et al., 2024b), though pre-defined criteria may not generalize well to real-world settings (Shankar et al., 2024). Our work builds upon CheckList (Ribeiro et al., 2020), which introduced structured behavioral testing for NLP models, TICK (Cook et al., 2024), which demonstrated decomposition benefits through model-generated criteria, and CheckEval (Lee et al., 2025), which showed that using a decomposition list of binary questions can effectively improve the average agreement across evaluator models and also reduce the score variance for text-generation tasks. We extend these approaches by training a dedicated scoring model that synthesizes multiple training signals, conducting broader evaluations across diverse benchmarks, and validating through human studies.

We have further discussion of how LMUNIT relates to recent work in Appendix A.4

2.5 Unified Evaluators

Recent work has focused on unifying different evaluation paradigms. DJPO (Wang et al., 2024b) improves human correlation by training LM judges through preference optimization (Rafailov et al., 2023), while Prometheus (Kim et al., 2024a,c) combines direct assessment and pairwise ranking capabilities through model weight merging. These approaches, along with fine-grained reward functions (Wu et al., 2023), show promise in both human and automatic evaluations.

LMUNIT extends these unified approaches while addressing limitations in interpretability, generalization, and fine-grained control. It decomposes evaluation into explicit testable criteria defined and refined by human experts, leveraging both LM judges (natural language understanding, flexible criteria) and reward models (precise scoring, preference learning) to enable reliable, interpretable, and actionable evaluation adaptable to diverse real-world requirements.

3 LMUNIT Methodology

To enable reliable scoring of natural language unit tests, we develop LMUNIT, a unified modeling approach that combines multi-objective training with natural language rationale generation. The key challenge lies in effectively integrating diverse training signals while maintaining both high accuracy and interpretable outputs. Here, we detail our approach to addressing this challenge through careful problem formulation, synthetic data generation, and our training methodology.

3.1 Problem Formulation

The core challenge in language model evaluation is developing scoring models that can reliably evaluate responses against specific criteria while providing interpretable reasoning. Our formulation centers on unit tests: given a unit test u, prompt p, and response r, we train models to generate both rationales and scores through the mapping $f(u, p, r) \rightarrow$ rationale, score.

Our approach builds on two existing forms of evaluation data: direct rating data $(p,r) \rightarrow$ score and preference data $(p,r_1,r_2) \rightarrow$ preference. We extend these into unit test-based formats:

1. Unit test direct data: $(u,\ p,\ r) \rightarrow$ score or $(u, p, r) \rightarrow$ rationale, score
2. Unit test preference data: $(u, p, r_1, r_2) \rightarrow$ pref or $(u, p, r_1, r_2) \rightarrow$ rationale$_1$, rationale$_2$, pref

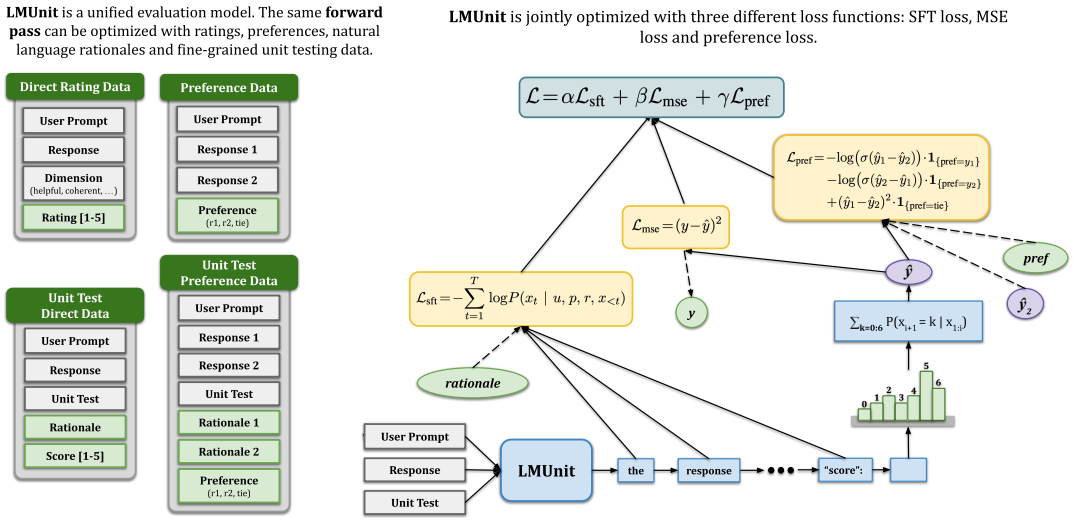

Figure 2: **LMUNIT Training Setup**: We leverage several different data sources (direct rating, preference, unit test direct, unit test preference) along with three different loss functions, to optimize the fine-grained scoring of LMUNIT.

This formulation leverages two complementary data sources: naturally occurring preference and rating data to capture human preferences and calibrate against absolute quality scales, alongside synthetic data that enables fine-grained evaluation of specific criteria with interpretable rationales. At inference time, LMUNIT can flexibly operate with or without rationale generation.

3.2 Synthetic Data Pipeline

Our data generation pipeline operationalizes the unit test formulation through three key stages, producing examples scored on a 1-5 scale where higher scores indicate better satisfaction of the criteria:

1. **Unit Test Generation**: For each prompt, we generate diverse unit tests targeting fine-grained quality criteria. To encourage focus on response-specific details, we optionally provide one or two responses during generation. We also maintain a set of coarse-grained global tests (see Table 11 for details) to ensure broad coverage of general quality dimensions.

2. **Contrastive Response Generation**: For each (u, p, r) triplet, we generate contrastive responses that vary systematically in how well they satisfy the unit test criteria. This creates rich training signal for learning fine-grained quality distinctions.

3. **Rationale and Score Generation**: For a subset of examples, we generate chain-of-thought rationales that explicitly reason through the evaluation criteria. Each rationale concludes with a score that must align with any existing seed data scores to maintain consistency.

We seed our synthetic data pipeline with prompts, responses, tests and scores from diverse sources including Nectar (Zhu et al., 2024), Prometheus (Kim et al., 2024a), Tulu3 (Lambert et al., 2024a), Complex Instructions (He et al., 2024), Infinity-Instruct (of Artificial Intelligence , BAAI), and HelpSteer2 (Wang et al., 2024d,c).

3.3 Training

LMUnit combines the strengths of generative judge models and classifier-based reward models through a unique multi-objective training approach. Given a unit test u, prompt p, and response r, the model outputs a sequence of rationale tokens $\text{rat} = (\text{rat}_1, ..., \text{r}_T)$ followed by a score token s. The probability distribution over possible score values $k \in 0,1,...,6$ is:

$$P(s = k \mid u, p, r, \text{rat}) = \text{softmax}(\mathbf{h}^T \mathbf{W}_s)_k \quad (1)$$

We compute a continuous score prediction through a weighted sum:

$$\hat{y} = \sum_{k=0}^{6} k \cdot P(s = k \mid u, p, r, \text{rat}) \quad (2)$$

The training objective combines three losses. First, SFT loss on the rationale and score tokens:

$$\mathcal{L}_{\text{sft}} = -\sum_{t=1}^{T} \log P(x_t \mid u, p, r, x_{<t}) \quad (3)$$

where $x_{1:t}$ represents tokens in both rationale and score sequences.

Second, MSE loss on the continuous score prediction:

$$\mathcal{L}_{\text{mse}} = (y - \hat{y})^2 \quad (4)$$

Third, preference loss:

$$\mathcal{L}_{\text{pref}} = -\log\big(\sigma(\hat{y}_1 - \hat{y}_2)\big) \cdot \mathbf{1}_{\{\text{pref}=y_1\}} - \log\big(\sigma(\hat{y}_2 - \hat{y}_1)\big) \cdot \mathbf{1}_{\{\text{pref}=y_2\}} + (\hat{y}_1 - \hat{y}_2)^2 \cdot \mathbf{1}_{\{\text{pref}=\text{tie}\}} \quad (5)$$

Here, σ is the sigmoid function. The final loss is a weighted combination:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{sft}} + \beta\mathcal{L}_{\text{mse}} + \gamma\mathcal{L}_{\text{pref}} \quad (6)$$

3.4 Post-Training of Rationales

While our initial model learns to generate rationales through imitation learning, there is no guarantee that these rationales actually improve scoring performance. We address this by collecting pairs of desirable and undesirable rationales for direct preference optimization (Rafailov et al., 2023), training the model to prefer rationales that lead to correct scoring. We employ several collection strategies: Through Refined, we collect on-policy rationales from our trained model and use the teacher to refine them through revisions (D'Oosterlinck et al., 2024) that improve scoring accuracy. With Harmonized, we provide the teacher with two model rationales from a preference pair to harmonize them with their samples' relative quality. In the Teacher-based strategy, we sample teacher rationales on known-score samples, using those with correct outcomes as chosen samples and incorrect ones as rejected. We compare these approaches in Table 4.

3.5 Bayesian Optimization of Global Unit Tests

Natural language unit tests decompose evaluation into fine-grained criteria through K global tests that assess dimensions like accuracy, safety, and coherence. The aggregation of these assessments into an overall score is crucial for valid evaluation. Rather than using standard uniform weighting, we learn optimal weights $w_1,...,w_K$ through Bayesian optimization over human preference data to maximize alignment between weighted test scores and human judgments. This process iteratively updates weights from a uniform initialization based on agreement with held-out human preferences.

4 Experiments

We conducted extensive experiments to evaluate LMUNIT and the natural language unit test paradigm. First, we evaluated the performance of LMUNIT on several evaluation benchmarks, comparing to LLMs as judges, reward models, and trained evaluation models. Next, we perform ablations to understand the impact of different methodologies, including loss functions and data mixture choices. Also, we examined improving rationales through post-training and analyzed the impact of decomposition through several unit test strategies. Finally, as shown in Appendix A.1, we also conducted two human subject studies to validate the advantages of the LMUNIT paradigms over LM judges

4.1 Experimental Setup

4.1.1 Model Configuration and Training Data

Our training data encompasses a diverse mix of preference judgments, direct scores, and rationales across multiple sources: (i) HELPSTEER 2 (50K pairs with ratings spanning five dimensions), (ii) PROMETHEUS (10K unpaired samples with ratings), (iii) SYNTH NON-RUBRIC (11K pairs with ratings and rationales), (iv) SYNTH RUBRIC (13K unpaired samples with ratings and rationales).

We train several variants of LMUNIT initialized from instruction-tuned LLaMa-3.1 models (8B, 70B). We train our models for 2000 steps using fixed weights (i.e., $\alpha = \beta = \gamma = 1$) for the different loss components, with a 5x loss multiplier applied to the rationale samples. The training uses the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 1e-6 and a cosine learning rate scheduler, using a batch size of 64 and a sequence length of 8K.

4.1.2 Evaluation Benchmarks

We evaluate our models on six benchmarks spanning diverse capabilities: Direct scores assessment (BigGenBench, Flask), Classification (Internal Unit Test set, Infobench), and preference evaluation (RewardBench , LFQA). At inference time, we compute a continuous score as the expected value of the possible scores in accordance to our training strategy described in Sec. 3.3. For dataset details, see Appendix A.2

4.2 Key Results

Our models demonstrate strong performance across diverse evaluation settings (Table 1). On direct assessment tasks, LMUNIT achieves state-of-the-art results with correlations of **72.03** on FLASK and **67.69** on BiGGen-Bench, where fine-grained evaluation is particularly important. In aggregate, LMUNIT achieves strong overall performance with scores of **79.74** (eight weighted global unit tests) and **79.29** (single unit test), outperforming general-purpose models like GPT-4 (78.29) and Claude-3.5 Sonnet (77.78). Even our smaller

Model	Direct Assessment		Classification		Pairwise Ranking		Average*
	Flask	BiGGen-Bench	Human-Internal	InfoBench	RewardBench	LFQA	
GPT-4o	69.00	65.00	81.80	**92.80**	84.60	76.54	77.59
Claude-3.5 Sonnet	67.25	61.83	84.53	91.58	84.23	**77.24**	76.43
Prometheus-2-7B	47.00	50.00	75.58	48.60	72.0	72.31	57.98
Prometheus-2-8x7B	54.00	52.00	77.82	87.85	74.5	74.23	68.52
Prometheus-2-BGB-8x7B	31.00	44.00	78.57	83.87	68.3	71.54	59.74
Llama-3-OffsetBias-8B	29.00	21.00	68.15	72.15	84.0	63.08	53.85
Skywork-Critic-Llama-3.1-8B	-	-	-	-	89.0	64.23	-
SFR-LLaMA-3.1-8B-Judge	52.00	59.00	-	**92.80**	88.7	68.85	72.27
SFR-LLaMA-3.1-70B-Judge	66.00	65.00	-	92.58	92.7	75.00	78.26
LMUNIT$_{\text{LLaMA3.1-8B}}$	60.02	64.46	**94.14**	91.26	83.23	71.54	74.10
LMUNIT$_{\text{LLaMA3.1-70B}}$	**72.03**	**67.69**	93.63	89.00	91.56	76.15	79.29
LMUNIT$_{\text{LLaMA3.1-70B}-Decomposed}$	**72.03**	**67.69**	93.63	89.00	90.54	74.62	78.78
LMUNIT$_{\text{LLaMA3.1-70B}-Decomposed-Weighted}$ †	**72.03**	**67.69**	93.63	89.00	**93.45**	76.53	**79.74**

Table 1: **Comprehensive Model Performance Comparison**: Evaluation results across multiple benchmarks showing model performance on various tasks. Metrics: (i) Pearson correlation coefficient for direct assessment, (ii) binary accuracy for classification tasks, and (iii) pairwise preference accuracy for pairwise comparisons. † represents our result with Bayesian optimization over pairwise benchmarks for learning global unit test weights, as described in Section 3.5. We learned dataset-level weights for LFQA and section-level weights for RewardBench by optimizing over model predictions on a 50% split of the dataset, following prior work (Wang et al., 2024d). We only apply the decomposed unit tests and weight optimization for RewardBench and LFQA since they lack fine-grained criteria for evaluation. We confirm that this technique generalizes to a held-out split of RewardBench in Table 15. Note that the Average column excludes Human-Internal scores in order to compare fairly against the non-public SFR-LLaMA baselines (as of December 2024).

LMUNIT$_{\text{LLaMA3.1-8B}}$ variant remains highly competitive with a 74.10 average score. For pairwise ranking tasks, using unweighted global unit tests slightly decreases overall performance to 78.78 (-**0.96**), but LMUNIT remains stronger than all other baselines. We recover this minor performance loss through Bayesian optimization of the global unit test weights while reaching **93.45** on RewardBench (+**2.91**) - though we note this weighting is learned on a subset of RewardBench itself, analogous to tuning hyperparameters on the test set (following a similar experimental setup as Wang et al. (2024d)). A more rigorous analysis using a proper held-out evaluation set is provided in Section 4.3.4, confirming the generalization of this method. These strong results across direct assessment, classification, and pairwise ranking tasks validate the effectiveness of our synthetic data pipeline, training setup, and unified scoring methodology, establishing LMUNIT as a state-of-the-art model for reliable evaluation.

4.3 Ablation Studies

We conduct extensive ablation studies to understand the key components driving LMUNIT's performance. Our analysis focuses on three main aspects: (1) the impact of different training objectives and data mixture compositions, (2) the role of rationales in model performance, and (3) strategies for unit test decomposition and aggregation. Additionally, we perform supplementary ablations on LMUNIT such as base-model architecture (A.3) unit test composition (A.3.2), Bayesian optimization with different models (3.5), and LMUNIT weighted inference (A.3.3).

4.3.1 Impact of Loss Functions

Our ablation studies in Table 2 demonstrate that combining training objectives (SFT, MSE, and preference loss) yields measurable improvements across our evaluation benchmarks (+0.5). LMUNIT-8B shows particularly significant gains on fine-grained evaluation datasets—9% on FLASK and 6% on BigGenbench—with more modest improvements (1-3%) on pairwise datasets that assess coarser-grained capabilities. These differential gains suggest our multi-objective approach is especially beneficial when evaluating nuanced LLM capabilities and when parametric capacity is limited, as evidenced by smaller improvements (+3%) at the 70B parameter scale.

4.3.2 Data Mixture Effects

We analyze how different compositions of training data affect LMUNIT's performance to identify the most effective mixture for robust evaluation capabilities. As shown in Table 3, rubric data is essential for strong performance on fine-grained direct assessment and that our synthetic data pipeline provides dramatic performance gains (+**3.52**) when synthetic rubric data is incorporated. We also observe that non-rubric synthetic data is most effective as preference pairs (+**4.04**) rather than direct scoring data (-**2.75**), likely due to the improved contrastive signal.

Training Loss	Direct Assessment		Classification		PairWise Ranking		Average
	Flask	BiGGen-Bench	Human-Internal	InfoBench	RewardBench	LFQA	
LMUNIT$_{\text{LLaMA3.1-8B}}$							
SFT	51.31	59.12	94.19	90.29	83.56	68.85	74.55
SFT + MSE	60.46	63.94	94.29	92.92	83.44	71.54	**77.77**
SFT + MSE + PREF	60.02	64.46	94.14	91.26	83.23	71.54	77.44
LMUNIT$_{\text{LLaMA3.1-70B}}$							
SFT	69.09	67.14	93.88	90.83	89.98	76.15	81.18
SFT + MSE	70.25	67.34	93.73	87.59	91.03	75.77	80.95
SFT + MSE + PREF	72.03	67.69	93.63	89.00	91.56	76.15	**81.68**

Table 2: **Training Loss Ablation Results**: Adding SFT, MSE, and preference loss components each contribute modest but consistent improvements to LMUNIT's performance across direct assessment (Pearson correlation), classification (binary accuracy), and pairwise ranking (preference accuracy) tasks.

Data Mix	Direct Assessment		Classification		PairWise Ranking		Average
	Flask	BiGGen-Bench	Human-Internal	InfoBench	RewardBench	LFQA	
Direct only							
HS2	57.0	42.26	94.74	88.60	91.31	69.23	73.86
HS2 + SYNTH NON-RUBRIC	47.00	42.00	93.83	88.80	86.00	69.00	71.11
HS2 + PROMETHEUS	64.90	59.27	93.43	87.50	91.40	71.15	77.94
HS2+ PROMETHEUS + SYNTH RUBRIC	71.60	67.94	94.89	89.19	91.70	73.50	**81.46**
Preference only							
SYNTH NON-RUBRIC	65.94	62.80	92.37	91.69	80.73	66.92	76.74
HS2	59.26	44.00	94.19	87.49	90.54	69.62	74.18
HS2 + SYNTH NON-RUBRIC	64.89	62.13	93.88	87.70	91.49	69.23	**78.22**
Full Data Mix							
ALL	72.03	67.69	93.63	89.00	91.56	76.15	**81.68**

Table 3: **Training Data Mix Ablations**: Our direct-only synthetic mix with rubrics dramatically improves model performance over baselines trained on open-source data only. Our synthetic preference data also strongly improves performance even without rubrics, likely due to fine-grained contrastive signal. Training on our full data mix yields our SOTA LMUNIT model. All models are initialized with Llama-3.1-70B. HS2 refers to HelpSteer2.

4.3.3 Impact of Rationales

Moving beyond simple imitation learning of rationales, we examine strategies to optimize rationale generation for better evaluation. As shown in Table 4, training with rationales improves model performance even when rationales are not used at test time (+0.2). While including rationales during inference initially leads to lower scores, our post-training optimization through DPO helps recover performance, with teacher-based pairs providing the largest gains (+1.1).

Training Process	Rationales?		Benchmarks			
	Train	Test	RewardBench	BigGenBench	Flask	**Avg**
LMUNIT Losses	✗	✗	91.1	67.4	**72.1**	76.9
LMUNIT Losses	✓	✗	**91.6**	**67.7**	72.0	**77.1**
LMUNIT Losses	✓	✓	83.8	62.1	64.2	70.0
LMUNIT Losses + DPO (H)	✓	✓	84.4	62.0	64.6	70.4
LMUNIT Losses + DPO (R)	✓	✓	84.2	61.8	65.0	70.3
LMUNIT Losses + DPO (T)	✓	✓	85.4	63.1	64.9	71.1

Table 4: Rationale Ablations: Training on rationale data improves LMUNIT$_{\text{LLaMA3.1-70B}}$ performance without test-time rationales, but test-time rationale generation decreases performance. DPO post-training improves rationale generation further.

4.3.4 Unit Test Decomposition Analysis

Our experiments with different unit test strategies on RewardBench (Table 15) reveal two key findings. First, global-level tests significantly outperform query-level tests across all categories, with section-level learned weights achieving the strongest results (+2.4 over unweighted aggregation). Second, the performance of fine-grained query-level tests degrades substantially, particularly on harder examples, though this can be partially mitigated by placing greater weight on earlier tests (+1.5).

These results highlight both the promise and challenges of our approach: while global unit tests provide a robust foundation for evaluation, developing effective fine-grained testing criteria remains difficult. The success of weighted global unit tests, coupled with the challenges of query-level decomposition, suggests an important direction for future work in developing more sophisticated test generation and aggregation strategies. Additional details of how decomposition is applied with

Bayesian optimization and with different base models can be seen at A.3.5

5 Discussion

Our experiments and analyses reveal several key insights about the effectiveness of our unit test-based evaluation framework and highlight important directions for future work:

LMUNIT Shows Benefits of Unified Training: Our empirical results validate the benefits of a unified scoring approach through three key findings: combining multiple training objectives improves performance across all evaluation settings (Table 2), incorporating diverse data types enhances model capabilities (Table 3), and LMUNIT's approach achieves state-of-the-art results on fine-grained evaluation benchmarks like FLASK and BiGGen-Bench (Table 1). These results suggest significant untapped potential in synthesizing different sources of evaluation signal – from human preferences and ratings to targeted synthetic data – particularly for fine-grained assessment tasks.

Unit Tests Enable Rich Human-in-the-Loop Evaluation: Language model evaluation frameworks should enable precise human steering while reducing noise and manual effort. Our results show this paradigm achieves both goals: structured criteria dramatically improve evaluation consistency and inter-annotator agreement (Figure 4), while offering multiple meaningful intervention points. Humans can write or refine test criteria, optimize test weights (Table 15), and guide development through decomposed feedback - leading to significantly more detailed error analysis in practice (Appendix A.1). This suggests unit tests can enable deeper, more reliable human-AI collaboration in evaluation.

Rationale Post-Training Improves Task Performance: A fundamental challenge in language models is developing genuine reasoning capabilities rather than simply learning to imitate human-like explanations. While training models to generate rationales through supervised learning can produce plausible-sounding explanations, this doesn't necessarily improve their underlying capabilities. Our work demonstrates two key insights about moving beyond imitation: first, training with rationales improves model performance even when not generating them at inference time (Table 4), and second, post-training optimization of rationales for task performance rather than imitation leads to further gains. This suggests a promising direction for developing better reasoning capabilities: using rationales not just as outputs to mimic but as a trainable intermediate step that can improve task performance while maintaining interpretability and enabling human feedback. Beyond LMUNIT, this approach can be extended to improve general-purpose model reasoning by optimizing rationales for downstream task performance rather than merely imitating ground-truth rationales.

Query-Level Unit Test Creation Remains Challenging: While our work advanced scoring and evaluation methodology, generating effective query-specific unit tests proved difficult. Global-level unit tests with learned weights significantly outperform query-level unit tests (Table 15), highlighting the need for better test generation approaches. Future work should explore end-to-end training of test generation, evaluate human-created tests at scale, and investigate when fine-grained decomposition justifies its complexity. These findings collectively point to both the promise and challenges of the unit testing paradigm for language model evaluation. The strong performance of LMUNIT demonstrates the potential of unified training approaches, while our human studies show how structured evaluation can enable more reliable and meaningful human oversight. Though challenges remain in test generation and optimal decomposition strategies, our results suggest this paradigm offers a practical path toward more reliable, interpretable, and human-aligned evaluation of language models.

6 Conclusion

This paper introduces natural language unit tests, a paradigm for language model evaluation that enables precise assessment through explicit, testable criteria. To implement this paradigm effectively, we develop LMUNIT, a unified scoring model that combines multi-objective training across preferences, direct ratings, and natural language rationales to achieve state-of-the-art performance on major evaluation benchmarks. Our results validate both the broader paradigm of decomposed evaluation and our novel scoring methodology. Looking ahead, this work opens several promising research directions: deeper integration of human feedback loops, enhanced scoring models with improved reasoning capabilities, and end-to-end training of unit test generation and scoring.

7 Limitations

LMUNIT shows promising results across multiple evaluation settings, though some shortcomings remain that provide potential research directions. The generation of query-specific unit tests, while functional, could benefit from more sophisticated approaches to better capture fine-grained evaluation criteria. The framework's reliance on human expertise for creating high-quality domain-specific unit tests, while valuable for ensuring evaluation quality, suggests opportunities for developing more automated test generation methods. Additionally, our synthetic data pipeline, which leverages existing datasets and language models for data generation, may inherit distributional biases that could influence evaluation outcomes. Although our results demonstrate strong performance despite these constraints, future work exploring automated test generation, reduced reliance on human expertise, and bias mitigation techniques could further enhance the framework's capabilities.

References

Anthropic. 2024. Claude technical report. Https://www.anthropic.com/news/claude-3-family.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A General Language Assistant as a Laboratory for Alignment. *ArXiv Preprint arXiv:2112.00861*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. 2024. RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs.

Bei Chen, Fengji Zhang, Anh Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, and Weizhu Chen. 2022. Codet: Code generation with generated tests. *arXiv preprint arXiv:2207.10397*.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the Judge? A Study on Judgement Bias. *ArXiv Preprint arXiv:2402.10669*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can Large Language Models Be an Alternative to Human Evaluations? *In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. *Preprint*, arXiv:1706.03741.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. *Advances in Neural Information Processing Systems*, 30.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. 2021. All That's 'Human' is Not Gold: Evaluating Human Evaluation of Generated Text. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. 2024. TICKing All the Boxes: Generated Checklists Improve LLM Evaluation and Generation.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on Evaluation Methods for Dialogue Systems. *Artificial Intelligence Review*, 54:755–810.

Karel D'Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri. 2024. Anchored Preference Optimization and Contrastive Revisions: Addressing Underspecification in Alignment. *ArXiv Preprint arXiv:2408.06266*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024a. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. *ArXiv Preprint arXiv:2404.04475*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024b. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. *Advances in Neural Information Processing Systems*, 36.

Jacob Eisenstein, Jonathan Berant, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D'Amour, Krishnamurthy Dj Dvijotham, Katherine A Heller, Stephen Robert Pfohl, and Deepak Ramachandran. 2023. Reward Model Underspecification in Language Model Alignment. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAs: Automated Evaluation of Retrieval Augmented Generation. *ArXiv Preprint arXiv:2309.15217*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as You Desire. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. 2024. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. *ArXiv Preprint arXiv:2404.15846*.

Tom Hosking, Phil Blunsom, and Max Bartolo. 2023. Human feedback is not gold standard. *ArXiv Preprint arXiv:2309.16349*.

David M Howcroft, Anya Belz, Miruna Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saad Mahamood, Simon Mille, Emiel Van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions. In *13th International Conference on Natural Language Generation 2020*, pages 169–182. Association for Computational Linguistics.

Jessica Huynh, Jeffrey Bigham, and Maxine Eskenazi. 2021. A Survey of NLP-Related Crowdsourcing Hits: What Works and What Does Not. *ArXiv Preprint arXiv:2111.05241*.

Karen Sparck Jones and Julia R Galliers. 1995. Evaluating natural language processing systems: An analysis and review.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The Perils of Using Mechanical Turk to Evaluate Open-Ended Text Generation.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing Fine-grained Evaluation Capability in Language Models.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. The BiGGen Bench: A Principled Benchmark for Fine-grained Evaluation of Language Models with Language Models.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024c. Prometheus 2: An open source language model specialized in evaluating other language models.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Min Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. *ArXiv Preprint arXiv:2305.13735*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *ArXiv Preprint arXiv:2302.14520*.

Dahyun Koo, Yejin Choi, and Eunsol Choi. 2023. Cognitive Biases in Large Language Models as Evaluators. *ArXiv Preprint arXiv:2312.05441*.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. Longeval: Guidelines for human evaluation of faithfulness in long-form summarization. *ArXiv Preprint arXiv:2301.13298*.

Nathan Lambert and Roberto Calandra. 2023. The alignment ceiling: Objective mismatch in reinforcement learning from human feedback. *ArXiv Preprint arXiv:2311.00168*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024a. TÜLU 3: Pushing Frontiers in Open Language Model Post-Training. *ArXiv Preprint arXiv:2411.15124*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024b. RewardBench: Evaluating Reward Models for Language Modeling.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025. Checkeval: A reliable llm-as-a-judge framework for evaluating text generation using checklists. *Preprint*, arXiv:2403.18771.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2024a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *Preprint*, arXiv:2411.16594.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *Preprint*, arXiv:2310.05470.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024b. Leveraging Large Language Models for NLG Evaluation: Advances and Challenges. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16028–16045.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024a. WILDBENCH: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild. *ArXiv Preprint arXiv:2406.04770*.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024b. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *Preprint*, arXiv:2406.04770.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *ArXiv Preprint arXiv:2305.13711*.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *ArXiv Preprint arXiv:1603.08023*.

Chris Yuhao Liu and Liang Zeng. 2024. Skywork Reward Model Series. `https://huggingface.co/Skywork`.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *ArXiv Preprint arXiv:2303.16634*.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024. Hd-eval: Aligning large language model evaluators through hierarchical criteria decomposition. *Preprint*, arXiv:2402.15754.

Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. *ArXiv Preprint arXiv:1708.07149*.

Shikib Mehri and Maxine Eskenazi. 2020a. Unsupervised evaluation of interactive dialog with dialogpt. *ArXiv Preprint arXiv:2006.12719*.

Shikib Mehri and Maxine Eskenazi. 2020b. USR: An Unsupervised and Reference Free Evaluation Metric for Dialog Generation. *ArXiv Preprint arXiv:2005.00456*.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. Rankme: Reliable human ratings for natural language generation. *ArXiv Preprint arXiv:1803.05928*.

Beijing Academy of Artificial Intelligence (BAAI). 2024. Infinity instruct. *ArXiv Preprint arXiv:2406.XXXX*.

OpenAI. 2023. GPT-4 Technical Report. *ArXiv Preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Qian Pan, Zahra Ashktorab, Michael Desmond, Martín Santillán Cooper, James Johnson, Rahul Nair, Elizabeth Daly, and Werner Geyer. 2024. Human-Centered Design Recommendations for LLM-as-a-judge. In *Proceedings of the 1st Human-Centered Large Language Modeling Workshop*, pages 16–29.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. *ArXiv Preprint arXiv:2401.03601*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. 2024. Lynx: An Open Source Hallucination Evaluation Model.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Paul Rottger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two Contrasting Data Annotation Paradigms for Subjective NLP Tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 175–190, Seattle, United States. Association for Computational Linguistics.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. *ArXiv Preprint arXiv:2311.09476*.

Jon Saad-Falcon, Adrian Gamarra Lafuente, Shlok Natarajan, Nahum Maru, Hristo Todorov, Etash Guha, E Kelly Buchanan, Mayee Chen, Neel Guha, Christopher Ré, and Azalia Mirhoseini. 2024. Archon: An architecture search framework for inference-time techniques. *arXiv preprint arXiv:2409.15254*.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2024. Branch-Solve-Merge Improves Large Language Model Evaluation and Generation.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. 2025. Learning to plan reason for evaluation with thinking-llm-as-a-judge. *Preprint*, arXiv:2501.18099.

Stephanie Schoch, Diyi Yang, and Yangfeng Ji. 2020. "This is a Problem, Don't You Agree?" Framing and Bias in Human Evaluation for Natural Language Generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pages 10–16.

Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *ArXiv Preprint arXiv:2310.03716*.

Eric Michael Smith, Orion Hsu, Rebecca Qian, Stephen Roller, Y-Lan Boureau, and Jason Weston. 2022. Human Evaluation of Conversations is an Open Problem: Comparing the Sensitivity of Various Methods for Evaluating Dialogue Agents. *ArXiv Preprint arXiv:2201.04723*.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 341–351. Springer.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv Preprint arXiv:2307.09288*.

Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering*, 6(3-4):363–377.

Marilyn A Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30:413–456.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun Zhao, Xiao Wang, Tao Ji, Hang Yan, Lixing Shen, Zhan Chen, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024a. Secrets of RLHF in Large Language Models Part II: Reward Modeling.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is ChatGPT a good NLG Evaluator? A Preliminary Study. *ArXiv Preprint arXiv:2303.04048*.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024b. Direct Judgement Preference Optimization. *ArXiv Preprint arXiv:2409.14664*.

Yuhang Wang, Ruochen Xu, Zhenrui Xu, Yihuai Jiang, Zhen Xu, and Wenpeng Yin. 2023b. Large Language Models are Inconsistent and Biased Evaluators. *ArXiv Preprint arXiv:2311.16881*.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2024c. HelpSteer2-Preference: Complementing Ratings with Preferences.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024d. HelpSteer2: Open-Source Dataset for Training Top-Performing Reward Models. *ArXiv Preprint arXiv:2406.08673*.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2023c. HelpSteer: Multi-Attribute Helpfulness Dataset for SteerLM.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-Grained Human Feedback Gives Better Rewards for Language Model Training. *Advances in Neural Information Processing Systems*, 36:59008–59033.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. WizardLM: Empowering Large Language Models to Follow Complex Instructions.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023b. A critical evaluation of evaluations for long-form question answering. *Preprint*, arXiv:2305.18201.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023. FLASK: Fine-grained Language Model Evaluation based on Alignment Skill Sets.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Zhiqiang Yuan, Yiling Lou, Mingwei Liu, Shiji Ding, Kaixin Wang, Yixuan Chen, and Xin Peng. 2023. No more manual tests? evaluating and improving chatgpt for unit test generation. *arXiv preprint arXiv:2305.04207*.

Chen Zhang, Luis Fernando D'Haro, Yiming Chen, Malu Zhang, and Haizhou Li. 2024a. A Comprehensive Analysis of the Effectiveness of Large Language Models as Automatic Dialogue Evaluators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19515–19524.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024b. Generative Verifiers: Reward Modeling as Next-Token Prediction.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv Preprint arXiv:1904.09675*.

Lianmin Zheng, Dacheng Xu, Jiajun Dong, Andy Zeng, Shuo Xie, Eric P Xing, and Percy Liang. 2023. Evaluation Biases for Large Language Models. *ArXiv Preprint arXiv:2305.17926*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *Conference on Empirical Methods in Natural Language Processing*.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. 2024. Starling-7b: Improving Helpfulness and Harmlessness with RLAIF. In *First Conference on Language Modeling*.

A Appendix

A.1 LMUNIT Human Subject Studies

We conducted two studies to validate key claims about natural language unit tests: (1) Whether this paradigm, implemented through LMUNIT, provides concrete advantages over traditional LM judges for developers working on real systems, and (2) Whether decomposing evaluation into explicit criteria can improve the quality of human preference data.

A.1.1 Case Study with LLM Developers

To evaluate whether decomposed evaluation helps developers better understand and improve language models, we conducted a controlled study with 16 researchers and engineers from NLP labs, covering domains in finance, publishing, software, and hardware development. The surveyed individuals utilized LMUNIT models over the course of 1-2 days, continuing their original evaluation workflows while comparing LMUNIT with traditional "LLM as a Judge" approaches. These researchers regularly develop LLM systems that integrate 70B+ parameter models with retrieval systems, frequently undergoing additional instruction fine-tuning and preference alignment datasets. When comparing evaluation approaches, LMUNIT enabled substantially more detailed analysis: participants identified **157%** more response attributes (10.8 vs 4.2) and **131%** more error modes (7.4 vs 3.2), rating both as significantly more important than those found through LM judges. These demands necessitated the development of reliable evaluation systems for understanding **1)** error modes of existing systems and **2)** actionable steps for improving existing approaches.

The insights provided by LMUNIT proved instrumental for improving both RAG systems and LLM systems more generally. 13 out of the 16 researchers surveyed stated that LMUNIT *helped them identify current error modes in their training pipelines*, inspiring them to make data selection and preprocessing decisions to address the failures directly. Eight researchers also said LMUNIT sparked them to make training pipeline decisions surrounding hyperparameters, dataset weighting, and in-context learning. Furthermore, six researchers reported these decisions led to a *10+ point boost in evaluation performance for instruction-following and reasoning tasks*. Most importantly, 15 of the 16 researchers expressed interest in using unit test-based frameworks for building ML pipelines going forward, assuming they align with evaluation metrics and human preferences for instruction-following and reasoning tasks. For detailed analysis, we provide an overview of the annotation guidelines in Table 7, annotation row examples in Table 5, and completed annotations in Table 6.

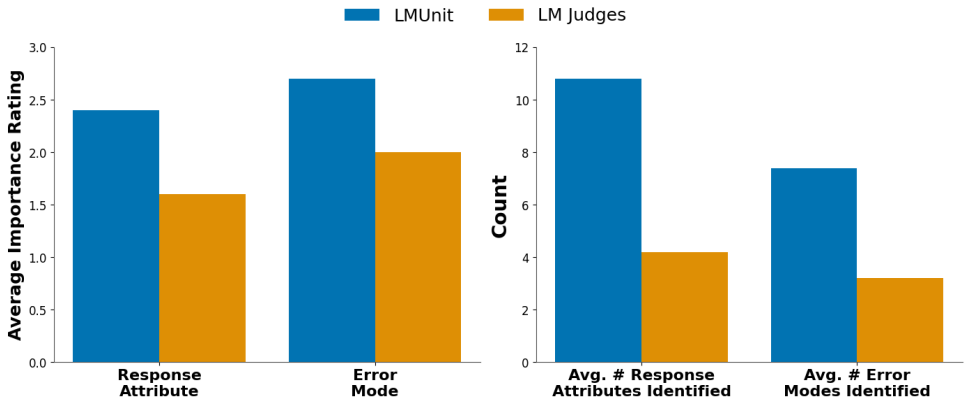

Figure 3: **LMUNIT Favored Over LM Judges for Identified Response Attributes and Error Modes**: LMUNIT is favored by LM researchers and engineers surveyed, allowing users to find greater numbers of important response attributes and error modes identified in LLM responses. Out of the 16 individuals surveyed, all of them favored LMUNIT over traditional LM judges.

We also gathered some illustrative anecdotes from study participants to reflect the benefits of unit test-based evaluation methods with LMUNIT:

- **Motivating LM System Decisions**: "We had suspected for a while that some of our training data was not diverse enough, but it was hard to prove with just LM judge feedback. The LMUNIT unit tests

	Instruction	Response #1	Response #2	LM Judge Verdict	LMUNIT Unit Tests with Verdicts for Better Response
Examples #1	How does the integration of healthcare analytics with electronic health records (EHRs) and the establishment of common technical standards contribute to improving patient care?	The integration of healthcare analytics with electronic health records (EHRs) and the establishment of common technical standards significantly contribute to improving patient care by providing a more coordinated, efficient, and data-driven approach to healthcare delivery...	**Integration of Healthcare Analytics with Electronic Health Records (EHRs)** * Enables the collection, aggregation, and analysis of vast amounts of clinical data from diverse sources, including EHRs, medical devices, and laboratory results. * Provides insights and analytics that help identify trends, predict outcomes, and improve patient care.	Response #1	- Does the response explain how healthcare analytics integration with EHRs impacts patient care? **[Passed]** - Does the response mention the role of common technical standards in improving interoperability? **[Passed]** - Does the response detail the major challenges addressed by the ONC's Federal Health IT Strategic Plan 2015-2020? **[Failed]** - Does the response cover how privacy concerns are addressed according to the ONC's plan? **[Passed]**
Examples #2	How might changes in the regulation and public perception of unmanned aircraft systems (UAS) influence the growth and application of urban drone technology?	The influence of changes in regulation and public perception of unmanned aircraft systems (UAS) on the growth and application of drone technology in urban environments over the next decade is significant and multifaceted...	**Regulations and Public Perception:** **1. Regulatory Landscape:** - Increased regulations governing airspace usage, safety, and privacy. - Stricter licensing and registration processes for drone operators...	Response #1	- Does the response adequately address how changes in regulation might impact the growth of UAS technology in urban environments? **[Passed]** - Does the response explain how changes in public perception might influence the application of UAS technology in urban environments? **[Passed]** - Does the response mention specific regulations that could affect UAS usage in urban areas? **[Passed]**

Table 5: **Comparison of LMUNIT vs. LM Judge for Human Feedback**

	Instruction	Response	Response Attributes	Error Modes
Example #1	How does the integration of healthcare analytics with electronic health records (EHRs) and the establishment of common technical standards contribute to improving patient care, while addressing the challenges related to privacy, data accuracy, and interoperability?	The integration of healthcare analytics with electronic health records (EHRs) and the establishment of common technical standards significantly contribute to improving patient care by providing a more coordinated, efficient, and data-driven approach to healthcare delivery. Several key strategies and challenges need to be addressed to achieve these improvements while safeguarding privacy, data accuracy, and interoperability. Firstly, healthcare analytics...	Explanation of Healthcare Analytics Integration, Discussion of EHRs (Electronic Health Records), Description of Common Technical Standards, Addressing Privacy Concerns, Focus on Data Accuracy, Discussion of Interoperability, Reference to ONC's Federal Health IT Strategic Plan, Use of Evidence-Based Examples, Explanation of Cost Reduction in Healthcare, Description of Stakeholder Collaboration	Hallucination of Unrelated Information, Omission of Key Concepts, Redundancy in Explanation, Inconsistent Use of Terminology, Lack of Coherence between Sections, Incomplete Explanation of Critical Points, Overly Generalized Statements
Example #2	How might changes in the regulation and public perception of unmanned aircraft systems (UAS) influence the growth and application of drone technology in urban environments over the next decade?	The influence of changes in regulation and public perception of unmanned aircraft systems (UAS) on the growth and application of drone technology in urban environments over the next decade is significant and multifaceted. Regulatory changes could impose stricter licensing and airspace regulations, while public concerns about safety and privacy might limit drone use in specific areas.	Explanation of UAS Regulation, Reference to Safety Standards, Discussion of Operational Restrictions, Mention of Air Traffic Management Systems, Integration of Collision Avoidance Technologies, Privacy Concerns in Urban Environments, Public Perception and Security Issues, Transparency in Data Capture and Processing	Failure to Address Privacy Concerns, Overlooking Public Perception and Security Issues, Vague Discussion on Commercial Applications, Inconsistent Explanation of Regulatory Compliance, Inaccurate Reference to Urban Growth Impact, Failure to Mention Innovation Amidst Regulations

Table 6: **LMUNIT Case Study Responses with Annotation Results**

revealed that the model was performing better on certain types of queries (i.e. summarization and multi-hop queries) while creating generic answers for others (i.e. analysis and calculation queries). This led us to augment the dataset with more varied examples and improve our retrieval process, leading to a performance increase for the LM system overall."

- **High-Resolution Feedback**: "With LM judges, we would often get long-winded explanations that did not really explain the issue clearly, which made it hard to figure out what was going on. Sometimes the judge verdict did not align with the explanation at all! However, LMUNIT gave us clear Passed/Failed results with specific criteria, allowing us to know what went wrong and where to fix it."

- **Improved Annotator Alignment**: "For our project, we noticed a frustrating gap between LM judge evaluations and the feedback from our annotators. The LM judges would pass responses that skipped crucial reasoning steps as long as the final answer was correct but annotators rejected responses for lacking logical progression. After switching to LMUNIT, the alignment with the annotators improved significantly. LMUNIT unit tests flagged responses that missed intermediate steps, just like the annotators. This allowed us to retrain the model with more targeted feedback, leading to better performance in tasks requiring step-by-step reasoning and saving us time on annotations."

Instruction	Response #1	Response #2	LM Judge Verdict	LMUNIT Unit Tests + Verdicts for Response#1	LMUNIT Unit Tests + Verdicts for Response#2
{text}	{text}	{text}	{#1 or #2}	Bulleted Queries + Verdicts	Bulleted Queries + Verdicts

Table 7: **Information for Comparing LM Judge and LMUNIT**: Given the following information, annotators then provide the response attributes, error modes, and their importances identified by each evaluation approach. We provide annotated row examples in Table 5 and completed annotations in Table 6.

A.1.2 Reducing Noise in Human Evaluation

Human preference data is crucial for training reward models (Christiano et al., 2017; Askell et al., 2021). However, inter-annotator agreement is often low (Wang et al., 2024a), with annotators struggling to weigh different factors consistently and give reliable signal (Howcroft et al., 2020). Since reducing task ambiguity has been shown to help improve agreement (Novikova et al., 2018; Huynh et al., 2021; Rottger et al., 2022), we investigated the benefits of decomposing evaluation into explicit criteria. We conducted an experiment with 15 experienced annotators on expressing judgements with 20 queries, comparing three approaches: unstructured preference judgments (Control), standardized evaluation criteria (Specification), and unit test-based evaluation (Unit Test). The Control group selected their preferred response with no additional guidance. The Specification group assessed each response against a five-point quality specification before selecting their preferred response. For the Unit test group, four experienced annotators first used a Google Sheets interface to create 4-8 unit tests per query. These tests were designed to verify that model responses were both accurate and grounded in the retrieved documents. After this step, the Unit Test group was instructed to answer the gold-standard targeted unit tests before picking.

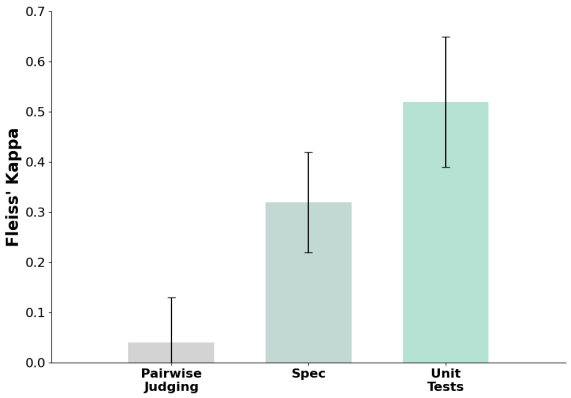

Figure 4: **LMUNIT Unit Test Scoring Improves Inter-Annotator Agreement on Preference Data**: Instructing annotators to answer gold-standard unit tests improves inter-annotated agreement by 48% and 20% compared to pairwise judging of responses or rubric-based scoring ("Spec"), respectively.

As shown in Figure 4 and in more detail in Table 8, the Control group showed low inter-annotator reliability (Fleiss' Kappa = 0.04), while the Unit Tests group achieved substantially higher agreement (Fleiss' Kappa = 0.52), demonstrating that structured decomposition significantly improves consistency in human evaluation. Annotators chose their preferred response after completing the unit tests and 89% of the time they selected the response with the largest number of satisfied unit tests. This further shows that answering unit tests guided their preference decisions.

	Agreement Overall	Kappa Overall	# Cases with 100% Agreement	# Queries with High Disagreement
Pairwise Judging	71%	0.04	3	12
Spec	80%	0.32	7	7
Unit Tests	86%	0.52	11	5

Table 8: **Unit Tests Improve Inter-Rater Agreement**: Unit test-based evaluation achieves substantially higher agreement rates and fewer cases of high disagreement compared to alternative approaches, such as pairwise judging and rubric-based scoring (i.e. "Spec"). High disagreement refers to queries in the 40-60% agreement range.

A.2 Evaluation Benchmarks Details

- **RewardBench** (Lambert et al., 2024b): A benchmark of pairwise model outputs across chat, reasoning, and safety domains. We measure agreement with human preference judgments.
- **LFQA** (Xu et al., 2023a): A benchmark of long-form question answering responses. We measure agreement with expert preference judgments.
- **BiGGen Bench** (Kim et al., 2024b): A comprehensive benchmark spanning 77 tasks across instruction-following, content refinement, grounding, and tool usage. We measure correlation with human assessment scores.
- **FLASK** (Ye et al., 2023): An evaluation framework covering 12 skills across logical thinking, knowledge application, problem handling, and user alignment. We measure correlation with human assessment scores.
- **InfoBench** (Qin et al., 2024): A collection of instruction-following tasks. Using the expert-validated split, we measure binary classification accuracy against expert consensus.
- **Internal Unit Test Set**: A targeted evaluation of 190 questions in the finance and engineering domains, with an average of five validated unit tests per question. We measure binary classification accuracy against human expert annotations.

A.3 Additional Ablations

A.3.1 Model Architechture

To validate LMUNIT with different base models, we trained it on LLaMA3.3-70b and Qwen2.5-72b. Our results in Table1 showed that LMUNIT consistently transforms these base models into strong evaluators across the benchmarks described in 4.1.2.

Model	Direct Assessment		Classification		Pairwise Ranking	
	Flask	BiGGen-Bench	Human-Internal	InfoBench	RewardBench	LFQA
$\text{LMUNIT}_{\text{LLaMa3.3-70b}}$	73.09	67.79	93.93	89.43	90.22	76.15
$\text{LMUNIT}_{\text{Qwen2.5-72b}}$	73.85	69.56	94.44	88.67	91.13	73.85

Table 9: **LMUNIT model ablations**: Evaluation results across multiple model variations. Results show that LMUNIT paradigm is applicable and effective to convert recent advancements of LLMs into strong evalutors

A.3.2 Unit-Test Composition

We evaluated how different information components from direct score benchmarks like Flask and BigGen-Bench contribute to improving the correlation between predicted scores and human ratings. These benchmarks provide three key elements: assessment questions, scoring rubrics (on a 1-5 scale), and reference answers. As shown in Table 10, incorporating additional information components incrementally improves the correlation with human ratings, with the combination of reference answers and rubrics yielding the strongest performance.

A.3.3 LMUNIT Inference

Inference Budget comparison: In our current setup, LMUNIT is computationally cheaper than our strongest baselines in 9. The strongest baselines such as SFR (Meng et al., 2024), Claude (Anthropic, 2024), and GPT-4o (OpenAI, 2023) were evaluated by generating CoT rationales – see the exact prompt in Appendix A of Meng et al. (2024). These models are all either equal in size or larger than LMUNIT. LMUNIT advances SoTA without the use of generated rationales, generating only a couple of tokens for each input to produce the output score. LMUNIT only introduces additional tokens in the input (linearly proportional to the number of unit tests), which is far less expensive than additional output tokens

Unit-Test Format	Direct Assessment	
	Flask	BiGGen-Bench
LMUNIT$_{\text{LLaMA3.1-8B}}$		
UNIT TEST QUESTION	58.35	56.47
UNIT TEST QUESTION + RUBRIC	58.20	61.56
UNIT TEST QUESTION + REFERENCE ANSWER	58.37	63.07
UNIT TEST QUESTION + RUBRIC + REFERENCE ANSWER	**60.02**	**64.46**
LMUNIT$_{\text{LLaMA3.1-70B}}$		
UNIT TEST QUESTION	67.20	61.01
UNIT TEST QUESTION + RUBRIC	65.76	66.39
UNIT TEST QUESTION + REFERENCE ANSWER	70.01	65.61
UNIT TEST QUESTION + RUBRIC + REFERENCE ANSWER	**72.03**	**67.69**

Table 10: **Unit-Test Composition Analysis**. We analyzed how the composition of unit tests affects model performance. We observed that enriching unit tests with detailed information, such as rubrics and reference answers, improves the correlation with human ratings.

Test ID	Unit Test
GUT-1	Is the response helpful and aligned with the spirit of what the prompt was asking for?
GUT-2	Does the response directly address the prompt's query or topic?
GUT-3	Are the facts and information presented in the response correct and reliable?
GUT-4	Is the response articulated in a clear and understandable manner?
GUT-5	Does the response provide a thorough answer, covering all aspects of the prompt?
GUT-6	Is the response succinct without omitting essential information?
GUT-7	Does the response maintain the reader's interest and encourage further thought or action?
GUT-8	Does the response adhere to ethical guidelines and avoid promoting harmful content?

Table 11: Global Unit Tests used for pairwise evaluations on RewardBench and LFQA

because input token processing is parallelized in modern systems. The roughly 8X increase in input tokens (assuming 8 unit tests) is strongly outweighed by the roughly 6-12X reduction in required output tokens (assuming CoT rationales are ~100-200 tokens, which is reasonable based on the examples shown in Appendix B of (Meng et al., 2024).

Weighted Score Inference: To analyze the impact of our weighted score inference, which consists of calculating the expected value over all possible score values $k \in \{0,1,...,6\}$, we conducted a comprehensive evaluation across various tasks. As demonstrated in Table 12, the weighted score approach—which aligns with our training methodology—yields an average performance improvement of 6% compared to the baseline method.

The performance gains vary by task type: classification and direct assessment tasks show approximately 3% improvement, while pairwise ranking tasks exhibit more substantial gains ranging from 6% to 20%.

From a computational efficiency perspective, our method only requires logprob calculations up to the 5th token (where the "score (k)" token appears), resulting in negligible computational overhead.

A.3.4 Rationale Quality

Rationale generation capabilities in LMUNIT can enhance model interpretability and help humans understand the scoring process, despite slightly degrading performance. To evaluate rationale quality, we

Inference Method	Direct Assessment		Classification		PairWise Ranking		Average
	Flask	BiGGen-Bench	Human-Internal	InfoBench	RewardBench	LFQA	
LMUNIT$_{\text{LLaMA3.1-70B}}$							
WEIGHTED SCORE	72.03	67.69	93.63	89.00	91.56	76.15	**81.68**
NOT-WEIGHTED SCORE	69.39	65.80	92.92	86.62	70.24*	68.46	**75.57**

Table 12: **Ablation of our weighted score inference**. Performance comparison of LMUNIT when calculating the expected value over all possible scores compared to greedy text-generation

compared LMUNIT with a strong, presumably larger model—Claude Sonnet 3.5. Our evaluation involved 400 randomly selected samples (200 from FLASK and 200 from BigGBench), using Sonnet 3.5 as an LLM evaluator to assess rationale quality on a 1-5 scale across three metrics:

- **Faithfulness**: Evaluates how faithful/well-correlated the rationale is corresponding to the score and rubric.
- **Coverage**: Evaluates how thoroughly the rationale covers all aspects of the evaluation criteria presented in the unit test and rubric.
- **Clarity**: Evaluate how logically consistent and well-structured the rationale is. A sensible and coherent rationale presents reasoning that flows naturally, avoids contradictions, maintains topical focus, and creates a unified explanation.

Table 13 shows that LMUNIT's rationales achieve 92% of Sonnet 3.5's quality, demonstrating strong interpretability potential. Despite a small quality gap, LMUNIT delivers high-quality rationales that effectively explain evaluation outcomes.

Metric	Sonnet 3.5	LMUNIT	Relative Performance
Faithfulness	4.87	4.40	90.3%
Coverage	4.72	4.23	89.6%
Clarity	4.48	4.31	96.2%

Table 13: **Rationale quality analysis**. Qualitative analysis of rationales generated by LMUNIT on Faithfulness, Coverage, and Clarity

A.3.5 Bayesian Optimization Details

Preference-Guided Weight Optimization: LLM applications are judged along several partially competing quality criteria (*helpfulness*, *faithfulness*, *style*, *safety* , *among others*), and humans implicitly assign different importance to each. Benchmarks that score one criterion at a time such as FLASK (Ye et al., 2023), BigGenBench (Kim et al., 2024b), Human-Internal, InfoBench (Qin et al., 2024) cannot reveal these trade-offs since the detailed unit tests are already present.

By contrast, RewardBench (Lambert et al., 2024b) and LFQA (Xu et al., 2023b) provide *pairwise* human-preference labels ("chosen" vs. "rejected" response) but do not expose the underlying criteria. We bridge this gap by introducing a set of $N = 8$ *global unit tests* (Table 11) and learning a global weight vector $\mathbf{w} \in [0,1]^N$ such that a weighted sum of unit-test scores best reproduces human choices. Because the objective is non-differentiable and comparatively cheap to evaluate, we cast weight learning as black-box optimisation and employ Bayesian Optimization (BO). The specific methodology that we use is the following:

1. We partition the collected pairwise preference data into disjoint development and test sets.
2. For each response r in the development set, we compute scores $s_i(r)$ across each of the N global unit tests, where $i \in \{1,2,...,N\}$.

3. We formulate an aggregation function $f(r)$ that computes a final score for each response as a weighted linear combination of its individual unit test scores:

$$f(r) = \sum_{i=1}^{N} w_i \cdot s_i(r) \tag{7}$$

where $w_i \in [0,1]$ are learnable weights shared across all samples. In our experimental setup, we utilize $N = 8$ global unit tests, resulting in 8 parameters to optimize.

4. We employ Bayesian optimization to iteratively refine the weight parameters $\{w_i\}_{i=1}^{N}$. Specifically, we maximize the probability that for each preference pair (r_c, r_r) where r_c is the chosen response and r_r is the rejected response, the aggregation function assigns a higher score to r_c than to r_r:

$$\max_{\{w_i\}_{i=1}^{N}} \mathbb{P}(f(r_c) > f(r_r)) \tag{8}$$

The optimization is conducted using the BayesianOptimization framework[1] with the Probability of Improvement acquisition function for 200 iterations and weight constraints $w_i \in [0,1]$.

5. We evaluate the performance of the learned weights on the held-out test set, measuring how frequently the aggregation function correctly ranks the chosen response higher than the rejected response.

Finally, it is worth noting that the learned weights are intended to be customized, reflecting the specific human preferences in that dataset. They are not intended to generalize to other settings.

Additional Bayesian Optimization Experiments: As described in Sec. 3.5, we performed Bayesian optimization method described in A.3.5 on our LMUNIT model to optimize the weights for unit tests in RewardBench. We compared our approach with the two strongest open-source baselines: Prometheus-2-8x7B and Prometheus-2-BGB-8x7B. Results demonstrate that while Bayesian optimization improves both Prometheus baselines, they still underperformed compared to $\text{LMUNIT}_{\text{LLaMA3.1-70B}}$. Notably, even the Bayesian-optimized Prometheus models failed to outperform the standard (non-optimized) LMUNIT. These findings suggest that LMUNIT's superior performance on Pairwise Ranking tasks stems primarily from its core characteristics—specifically its training strategy and data collection methodology—rather than from weight optimization techniques such as Bayesian optimization.

	RewardBench		LFQA	
Model	No-weighted	Bayes opt.	No-weighted	Bayes opt.
$\text{LMUNIT}_{\text{LLaMA3.1-70B}}$	**90.54**	**93.45**	**74.62**	**76.53**
prometheus-bgb-8x7b-v2.0	76.38	79.79	67.31	71.54
prometheus-8x7b-v2.0	80.49	89.06	71.54	72.30

Table 14: **Bayesian Optimization Ablation**: Peformance comparison between the two strongest open-source baseliens (Prometheus-2-8x7B, Prometheus-2-BGB-8x7B) and LMUNIT. LMUNIT outperforms both with and without Bayesian optimization, highlighting the effectiveness of our training strategy and data collection.

A.4 LMUNIT in Relation to Prior Approaches

Our paradigm extends beyond prior criteria-based evaluation approaches by unifying five axes of evaluation into a single framework, providing thorough ablations to demonstrate the contribution of each one.

1. Criterion type: Each unit test captures a distinct criterion.

2. Criterion granularity: Each unit test can be made more specific via the inclusion of more details, a rubric, and/or a reference answer.

3. Criterion importance: Each unit test is assigned an importance weight, which can either be specified by the user or learned directly from human preference data.

[1] https://github.com/bayesian-optimization/BayesianOptimization

4. Score granularity: Our evaluator has been explicitly trained to express fine-grained differences in quality through a continuous score (unlike discrete or binary scores produced by most generative judge models).

5. Natural language rationales: The interpretability of scores can be increased by enabling the generation of rationales while preserving granular (continuous) scoring ability.

Most prior papers in LLM criteria-based evaluation focus on either criterion type or criteria granularity. Checklist (Ribeiro et al., 2020) is an earlier work that extends NLP model evaluation beyond accuracy to multiple criteria (unit tests). While being a foundational contribution, the paper does not consider the other axes mentioned above. Branch-Merge-Solve (Saha et al., 2024) shows the advantages of varying criterion type, but the criteria and score granularity are limited because the judge is not given a rubric to score against and has not been explicitly trained to distinguish fine-grained differences. Furthermore, the "merge" step aggregates criterion scores without considering their importance. Auto-J (Li et al., 2023) also shows the advantages of expanding criterion type while criteria granularity is quite under-specified (see Table 17 of their paper) and criterion importance is not addressed. Prometheus 2 (Kim et al., 2024c) directly addresses criterion granularity with fine-grained, query-specific rubrics, but their results and analysis neglect criterion type and criterion importance. HDEval (Liu et al., 2024) provides a principled approach for criterion importance, but their approach is focused on optimizing for coarse-grained task-level performance evaluation for a small set of tasks. Their training process is not optimized to distinguish fine-grained differences for a given criterion (limiting score granularity), and they do not evaluate on fine-grained criteria benchmarks.

Our work expands LLM evaluation across all 5 axes above. We propose a novel approach to criterion importance, showing that we can directly learn the importance of arbitrary criteria at the global level via Bayesian optimization using pairwise preference data (Section 3.5). We also demonstrate gains from further score granularity via multi-loss optimization (Section 3.3) and test-time weighted scoring (Table 12).

Additional related work demonstrates consistent findings with our paper despite different goals. WildBench (Lin et al., 2024b) focuses on developing an effective benchmark with automated metrics, sharing a set of queries with human-curated query-level criteria leading to more reliable scoring, consistent with the more general natural language unit test paradigm we explore in this paper. Thinking-LLM-as-a-Judge (Saha et al., 2025) proposes a DPO-based recipe to refine rationales that lead to reliable task-level performance evaluation. While similar to our DPO rationale experiments, this work does not investigate other axes of evaluation, such as criterion importance or improved score granularity.

Technique	RewardBench Subset				
	Chat	Chat Hard	Safety	Reasoning	Average
Global-Level Unit Tests					
Single Test	96.1	86.0	92.7	91.6	91.6
Unweighted Tests	97.2	79.9	93.2	93.4	91.0
Dataset-Level Learned Weights	95.6	84.3	93.2	95.7	92.2
Section-Level Learned Weights	**97.8**	**86.5**	**93.5**	**95.8**	**93.4**
Query-Level Unit Tests					
Single Test	92.8	78.6	84.1	83.7	84.8
Unweighted Tests	92.8	67.6	84.6	82.1	81.8
Exponentially Decaying Weights	93.9	72.9	84.9	81.4	83.3

Table 15: **Unit Test Decomposition**: RewardBench samples are scored using either 8 global tests (Table 11) or 8 query-specific tests generated by Claude-3.5-Sonnet. For learned weights, Bayesian optimization is applied to LMUNITLLaMA3.1-70B predictions on 50% of RewardBench. For decaying weights, each nth test is weighted by 0.8^n. Results reported on 50% held-out RewardBench data. Single test results use only the "Is the response helpful?" global test or first query-level test.