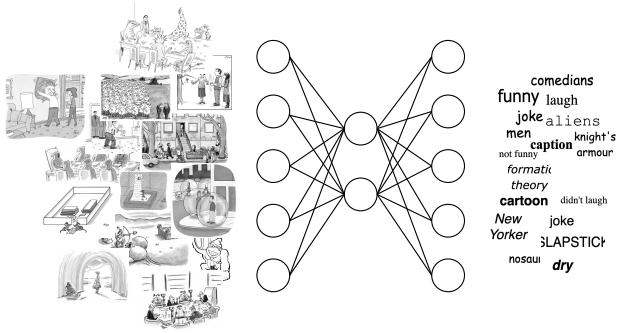
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG



Generating humor with information-theoretic reward signals

A system trained to write funny cartoon captions and humorous news headlines, optimized with reinforcement learning using human preference reward models and an information-theoretic approach to humor theory ☺

Master's thesis in Computer science and engineering

Carl Anton Lange

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2026

Master's thesis 2026

Generating humor with information-theoretic reward signals

A system trained to write funny cartoon captions and humorous news headlines, optimized with reinforcement learning using human preference reward models and an information-theoretic approach to humor theory ◡̈

Carl Anton Lange

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Generating humor with information-theoretic reward signals
A system trained to write funny cartoon captions and humorous news headlines, optimized with reinforcement learning using human preference reward models and an information-theoretic approach to humor theory ◡̈
Carl Anton Lange

Supervisor: Kivanç Tatar, Computer Science and Engineering
Examiner: Richard Johansson, Computer Science and Engineering

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2026

Abstract

Humor is ubiquitous in human communication, yet it remains difficult for large language models to both generate and evaluate. This thesis investigates whether information-theoretic reward signals, paired with a trained reward model, can align a language model to generate funny outputs. We frame humor generation as a reinforcement learning problem on two tasks: editing a single word of a news headline (Humicroedit) and writing captions for the New Yorker Cartoon Caption Contest (NYCCC), the latter serving as the main track. The training pipeline pairs a Bradley-Terry humor judge, trained on human preference data to score funniness, with the Context-based score for Value and Originality (CoVO), an information-theoretic reward that balances surprise against topical relevance and operationalizes incongruity-resolution theory. A Qwen3 generator is first fine-tuned on high-quality human examples and then optimized with Group Relative Policy Optimization (GRPO) against the combined reward. Across both tasks, the aligned generator does not produce funnier output than the fine-tuned baseline. Reward arms are either statistically indistinguishable from the SFT initialization or collapse into degenerate, reward-hacked text, and alternative algorithms (PPO, DPO) reproduce the same plateau. A series of ablations isolates the problem: the optimizer is healthy, but neither the humor judge nor the CoVO reward provides a usable gradient, as they track incongruity without registering whether it is resolved. The reward overfits to surface features, such as caption tone, rather than to humor. Generated humor lacks the alternative interpretation that resolves incongruity: the hidden or initially unexpected reading central to human humor perception. A human study of 920 ranking judgments confirms that participants strongly prefer human-written captions and do not rate GRPO output above the SFT baseline. We conclude that reliable humor generation and evaluation require world knowledge, cultural grounding, and reasoning about why something is funny, none of which are captured by caption-level preference labels or by information-theoretic proxies alone.

Keywords: Computational Humor, Reinforcement Learning, Information Theory, Large Language Models.

Acknowledgements

My friends and family know that humor has always been a big part of my life: creating, sharing, and laughing about jokes with others is among my favorite ways to connect with people. This thesis gave me the opportunity to dive deep into researching the mechanisms of humor. I combined this passion, which had been only a part of my private life, with my intellectual interest in Artificial Intelligence. Throughout the thesis project, I grew professionally and personally. For this, I would like to express my gratitude towards my supervisor, Kivanç Tatar, for guiding me through this research project. When I approached Kivanç with my idea of training a system for humor generation, he supported me from structured ideation, through research design, to the scientific evaluation of a topic as complex as humor. Richard Johansson, my examiner, provided me with very useful feedback and inspiration from his NLP background. Lastly, I want to thank my friends and family for enduring and listening to the (mostly unfunny) joke attempts my models made throughout the training process ◡̈. I received all the support that I could have ever asked for.

Carl Lange, Gothenburg, 2026-06-19

Contents

List of Figures

List of Tables

1

Introduction

> Laughter is an affection arising from the sudden transformation of a strained expectation into nothing.
>
> Kant [1790/ 1987]

Kant already realized that humor arises from the interaction between incongruent expectations. Humor is omnipresent and plays a major role in communication and social interaction. Whether we perceive a situation or statement as funny is highly subjective and depends on cognitive associations, social context, culture, personal preference, and linguistic structure. Thus, generating something humorous requires more than linguistic skill.

1.1 Humor Generation

As AI permeates everyday life, one may question whether humor-capable systems serve any purpose beyond short-lived personal entertainment. Recent works studied use cases where the humor capabilities of AI were not the primary goal but rather a means to an end. Kolomaznik et al. [2024] conducted a review of success criteria for integrating AI into daily tasks. Kolomaznik et al. found that users' productivity correlates with the system's emotional capabilities. In other words, human-AI collaboration relies on the AI's socio-emotional understanding of humans. Niculescu et al. [2013] found that humor plays a major role in this process, where humor and empathy modulated how much users enjoyed tasks that required robot interaction. Moreover, humans are more tolerant of failing AI systems in customer service applications when they respond humorously [Xu and Liu, 2022]. In other applications, humor-capable AI systems are used by humans in collaboration to create content. According to Kariyawasam et al. [2024], such a tool helped humans write funnier cartoon captions.

These are just some examples of how humor capabilities can make the interaction with AI systems more enjoyable. Beyond applications for directly generating humor, they warrant research into humor in AI and Large Language Models (LLMs). Although LLMs have demonstrated proficiency in many NLP tasks, humor-related tasks remain a challenge, including humor generation [Lemmens and De Marez, 2026, Amin and

Burghardt, 2020a, Hempelmann et al., 2025], joke detection [Hempelmann et al., 2025, Song et al., 2025], and humor understanding [Kalloniatis and Adamidis, 2024, Hessel et al., 2023]. One difficulty in the field of computational humor is evaluating results. Some studies rely on human evaluations of generated humor, which suffer from high inter-rater disagreement [Lemmens and De Marez, 2026]. Others use automated humor metrics, which often show a notoriously low correlation with human judgments or capture only one dimension of humor [Ismayilzada et al., 2024]. Existing datasets are diverse, and tasks range from binary humor classification (often manageable for models [Lemmens and De Marez, 2026]) to the much more challenging tasks of explaining *why* something is funny and generating novel humorous content. The surveys and literature reviews conclude that the multi-dimensionality of humor mandates neural approaches to 1) incorporate humor theories, 2) use formalisms that quantify different aspects of humor, and 3) use optimization frameworks that align LLMs with human preferences.

1.2 Aims

We frame humor generation as a reinforcement learning problem and design an AI system to solve it. Our approach adheres to all three recommendations: an LLM-based humor generation model is first aligned to human preference data, then optimized to maximize a composite reward that quantifies humor and incorporates humor theory. To quantify humor, we train a reward model on human ratings of humorous content. A second reward term operationalizes incongruity-resolution, a mechanism central to most humor theories, which describes how jokes set up expectations and parallel-but-conflicting interpretations, and then resolve them unexpectedly. To model incongruity-resolution, we use the Contextual Value-Originality framework (CoVO), which quantifies the trade-off between surprise and relevance in LLM outputs Franceschelli and Musolesi [2025].

Our system is trained for two tasks that require humor generation. The first task is news headline-editing on the Humicroedit dataset [Hossain et al., 2020a]. This task frames humor as changing a single word in a headline to turn it from serious to funny. This dataset has been a common benchmark for humor evaluation in recent years. Its narrow scope (changing precisely one word) allows us to study the behavior of the generation model well. The Humicroedit dataset contains continuous human-rated funniness scores for each edited news headline, which we use to train a reward model. To illustrate the task, consider the following example:

```
Original headline: President Trump's first year <anniversary/>
report card, with grades from A+ to F
Word to replace: anniversary
Human-written replacement word: Kindergarten
Human-graded funniness score: 3/3
```

Our system receives the original headline, chooses freely which word to replace, and then generates the replacement word.

Since the task is narrow and the dataset size is small, we apply our approach to another task: writing humorous captions for cartoons from the New Yorker Cartoon Captioning Contest (NYCCC). This dataset provides cartoon images, their text descriptions, and human-written cartoon captions, each rated with a continuous humor score. The large scale, high curation, and broader scope of this dataset make it an interesting comparison to Humicroedit, and we focus most efforts on the cartoon captioning task. In the following, we show a data sample from the NYCCC task in Figure 1. For this task, our system receives the textual cartoon description and the cartoon image and has the task to generate the caption.

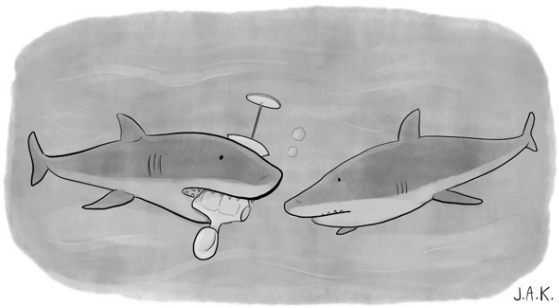


Figure 1: Cartoon #555.
Description: "Two sharks are talking underwater. One of them has a mannequin in its teeth. **Location:** the sea. **Entities:** Shark, Mannequin"
Human-written caption: "I had the tailor over for lunch"
Human-graded funniness score: 2.015

Although the two tasks have different formats, the overarching problem statement for our approach is the same. From human-written examples and their pairwise funniness preferences, we learn a surrogate humor judge $\mathcal{J}$. The generation problem is then: given a task input x, produce an output $\hat{y}$ that maximizes $\mathcal{J}(\hat{y})$ while balancing the surprise and relevance of $\hat{y}$, as quantified by the CoVO score. In the specific case of headline editing, the input is an original headline; the generator model selects a word to edit and produces the replacement word as output. For NYCCC, the model generates a cartoon caption from a textual cartoon description or a cartoon image.

1.3 Goals & Contributions

This thesis aims to align a language model to generate funnier outputs using a composite reward, and to test whether an information-theoretic signal can capture the incongruity resolution that humor depends on. We pursue four objectives

1. Train a learned humor reward: a Bradley-Terry judge $\mathcal{J}$ that infers relative funniness from human preference pairs, serving as a reward signal in both tasks.

2. Operationalize incongruity resolution as an information-theoretic reward by adopting the CoVO score, which separates a generation's originality (surprise) from its value (topical relevance), and investigate whether rewarding this trade-off tracks resolved incongruity rather than surprise alone.

3. Derive a vision-language extension of CoVO's value term via CLIP and establish its theoretical equivalence to the original formulation, so the reward can be applied to image-grounded humor tasks.

4. Combine the humor reward and the CoVO signal into a single reward, align a generator (Qwen3) with GRPO, and evaluate whether this improves funniness over a supervised fine-tuned baseline, using automated metrics and a human preference study.

These objectives help us to answer two central research questions:

RQ1 Does aligning the generator with a composite reward $\mathcal{J}$ + CoVO improve estimated funniness over a fine-tuned baseline?

RQ2 Is the learned humor judge $\mathcal{J}$ a reliable reward signal, and can it capture incongruity resolution?

Thesis outline. Chapter 2 summarizes the relevant literature on humor theory, computational creativity, prior LLM-based humor systems, and the information-theoretic framework we adopt. Chapter 3 describes the training pipeline and how it is applied to the two datasets. Chapter 4 reports component-level results, ablations isolating the failure modes, and the human user study. Chapter 5 answers the research questions and discusses the findings, while chapter 6 concludes the thesis and outlines directions for future work.

2

Background

This chapter grounds the thesis in six bodies of literature, in the order in which their concepts feed into our method. Section 2.1 reviews humor theory, with a focus on incongruity-resolution as the mechanism of our reward design. Section 2.2 extends this to computational creativity, where novelty, surprise, and value are operationalized in similar terms. Section 2.3 covers affective computing, motivating why surface emotion features alone do not capture humor. Section 2.4 introduces language-model alignment, the post-training paradigm we use to optimize the generator against a humor reward. Section 2.5 surveys prior LLM-based humor systems, identifying the persistent gap between theory and practice. Finally, Section 2.6 introduces the information-theoretic formalism we adopt to operationalize surprise and relevance as continuous reward signals.

2.1 Humor Theory

Understanding what makes something funny is a prerequisite for building systems that generate it. Before examining how computational systems approach humor, we first consider the theoretical perspectives on humor that have emerged from linguistics and psychology. Most effective computational humor systems operationalize these theories explicitly in the process of learning from language data. According to the incongruity-resolution (IR) theory, humor arises when one first notices an incompatibility or ambiguity in a statement, often called the *setup*, which is then resolved in the *punchline*. This theory has been described as early as the 1700s, according to Keith-Spiegel [1972]. It was formalized by Ritchie [1999], who distinguishes two cases of IR: surprise disambiguation, according to which humor arises when the punchline forces a reinterpretation of the ambiguous setup, and Suls [1972] two-stage model. In the first stage, one detects that the punchline violates the setup, triggering the second stage: the search for a "cognitive rule" that makes sense of the apparent violation. IR theory is used throughout many neural humor generation and evaluation systems Amin and Burghardt [2020b], Zhang et al. [2025]. When operationalized, incongruity is often modeled as the surprise or novelty of the information contained in the *setup*, followed by resolution that combines seemingly incongruent information. Other theories for humor, viewed as complementing each other rather than competing [Ferguson and Ford, 2008], include:

Benign Violation Theory attributes humor to situations where some social or cognitive norm is violated, but to an extent that is interpreted as benign [McGraw and Warren, 2010].

Superiority Theory frames humor as a sudden awareness of one's superiority over others' flaws, as outlined in Ferguson and Ford [2008].

Relief Theory is associated with Freud and Spencer. Here, humor acts as a psychological release mechanism for repressed emotions and nervous energy.

2.2 Computational Creativity

Generating humor is, at its core, a creative act. A system that merely recombines familiar joke templates is unlikely to produce genuine incongruity. This motivates a look at the broader field of computational creativity, which studies how machines can produce outputs that are novel, useful, and surprising, the same properties that humor theories identify as central to what makes something funny. Boden [2004] defines creativity as the ability to produce novel, useful, and surprising ideas. Ismayilzada et al. [2024] give an overview of recent advancements in computational creativity. They find that modern generative models excel at surface-level linguistic and artistic creativity, but struggle with deep creative processes such as abstraction, analogy, and creative problem-solving. In the context of humor generation, this suggests that LLMs can produce generic humor based on common joke patterns, but are likely to fail at the creative abstractions and deeper semantic connections required to create genuine incongruities Ismayilzada et al. [2024]. The authors also highlight that LLMs perform relatively well on *divergent thinking* tasks, where they have to come up with diverse ideas or solutions, but perform poorly during *convergent thinking* tests that demand one to decide on the best solution for a task, given a range of diverse options. Extrapolating this to humor, LLMs are likely good at generating several diverse ideas for a joke. Convergent thinking assumes an optimal choice. In humor, there usually is no "optimal" joke, as humor is highly contextual. Thus, an inability in convergent thinking could manifest as an inability to distinguish creative humor patterns from flat ones. It should be noted that the findings in Ismayilzada et al. [2024] are only summaries of benchmark- and task-specific LLM performance. Whether these generalize to untested tasks such as humor generation remains unclear. The authors also argue that reliable evaluation of computational creativity is a bottleneck, with humans providing subjective signals and automated metrics being one-dimensional.

Huang et al. [2025] introduce *LoTBench*, an evaluation framework for creativity, that measures LLMs in their ability to perform leap-of-thought (LoT) reasoning rather than the semantic similarity of the model's output and the target. The authors argue that simply comparing outputs neglects the reasoning process behind the solution, which allows for correct solutions with shallow reasoning behind it. This style of evaluation is relevant to humor generation, which often entails a disruption or incongruity, because it rewards creative leaps of thought in the humor generation process.

Ismayilzada et al. [2025] operationalize the different dimensions of creativity in a new alignment method for LLMs, Creative Preference Optimization (CrPO). They quantify novelty, surprise, diversity, and quality of LLM outputs and inject this into the loss function of Direct Preference Optimization (DPO, Rafailov et al. [2023]), a post-training method that uses preference pairs. Ismayilzada et al.'s results suggest that explicitly optimizing for multiple creativity dimensions within preference-learning methods produces more creative LLM outputs than prompting or standard alignment, without sacrificing quality.

The parallels between computational creativity and computational humor are structural. Both agree that a good output must be novel enough to surprise and relevant enough to remain coherent. In humor, this maps directly onto incongruity-resolution theory: the setup establishes relevance, the punchline introduces surprise, and the resolution requires a creative leap connecting them. This highlights what a good formalism must capture: not novelty or relevance in isolation, but their interaction. A punchline that is maximally surprising but semantically unrelated is nonsense; one that is predictable is not funny. The information-theoretic framework adopted in this thesis is designed precisely to navigate this trade-off.

2.3 Affective Computing

Humor does not work with just objective incongruities; these only work if they inflict some emotional response in the reader. This raises the question of how machines can detect and model emotion in text. The field of affective computing studies this challenge, and its methods and limitations are directly relevant to the understanding and evaluation of humor. In Wang et al. [2022]'s survey of the field, the authors distinguish discrete models that treat emotions as discrete categories (e.g., anger, disgust, fear) and dimensional or continuous models. Discrete models map words or text passages to one or more fixed emotions and take the form of simple lookup tables. Continuous models interpret emotions as continuous and multi-dimensional, asking, for example, how positive or negative, active or calming, or dominant versus submissive a feeling is. Such models are more expressive but require more compute, as state-of-the-art methods use deep learning [Wang et al., 2022]. The paper establishes that discrete models force emotion into fixed categories and that sentiment analysis typically yields only positive/negative/neutral outputs. Humor, which depends on incongruity and implicit meaning, cannot be adequately captured by either approach, as the paper itself identifies implicit emotion recognition (covering irony and sarcasm) as a key unsolved challenge for text-based methods.

2.4 Language Model Alignment

LLMs are the standard approach in humor-related tasks. These models have shown significant improvements over the past decade, attributable to a mix of theoretical advances in algorithms, the scaling of training data and compute, and engineering efforts. Importantly for this work, training LLMs is a multi-step process. Models

first undergo a so-called pretraining stage, in which they are trained to predict the next token of a text corpus in a supervised fashion. After this stage, the models are capable of writing high-quality prose and have some factual knowledge, which depends on the text corpora they were trained on. While this step has been common practice for some time, the boom in LLMs, especially chatbots and task-specific models, can be attributed to a secondary training step. After pretraining, models undergo post-training, during which they are aligned to answer in accordance with human preferences and (sometimes) trained on specific tasks. The literature on post-training is vast, and we will not go into detail here. Next, we introduce three post-training methods that are common practice when using LLMs for humor-related tasks. We first describe supervised fine-tuning, the simplest form of task adaptation, and then turn to two preference-based alternatives: reinforcement learning (PPO and GRPO), which optimizes against a learned reward model, and direct preference optimization, which removes the reward model and trains on preference pairs directly.

2.4.1 Supervised Fine-tuning

In this post-training step, the LLM is trained on a task-specific dataset containing instruction-response pairs Ouyang et al. [2022]. Supervised fine-tuning (SFT) typically teaches models to perform a specific task, such as summarizing a text. The model is then trained using a supervised cross-entropy loss on the predicted tokens, given a (text + summarization instruction, summary) pair.

2.4.2 Reinforcement Learning

SFT teaches task structure but cannot easily encode preferences that are only available as relative comparisons between outputs. Reinforcement learning addresses this gap by replacing the explicit target tokens of SFT with a scalar reward, allowing the model to learn from human judgements about whole responses rather than from token-level supervision.

Reinforcement Learning (RL) is a paradigm in which an objective or task is defined implicitly through a reward function. The reward reinforces positive behavior and punishes undesired behavior by assigning scalar values to model actions. The model is trained to maximize the reward. While RL has its origins in fields related to robotics and control, it has seen a renaissance in LLM post-training pipelines. Ouyang et al. [2022] used an RL algorithm to align a language model to human preferences. Specifically, they used human ratings of different versions of an LLM's output as a reward signal, such that outputs preferred by human raters received higher rewards, and optimized the LLM to maximize that reward. This is called Reinforcement Learning from Human Feedback (RLHF). Several RL algorithms have been used to align language models.

Proximal Policy Optimization (PPO, Schulman et al. [2017]) is widely regarded as one of the most successful deep RL algorithms due to its relatively simple implementation and stable performance on complex learning tasks.

Group Relative Policy Optimization (GRPO), introduced by Shao et al. [2024], has been widely adopted as the preferred RL algorithm for fine-tuning models. Its main differences from PPO are the absence of a value function network. Instead of estimating the expected advantage of a model's answer, GRPO computes advantages in a group-wise fashion: for a given prompt, k responses are generated, each receiving a reward r. Advantages A are defined relative to the other responses in the group:

$$A_i = \frac{r(\hat{y}_i) - \mu}{\sigma}, \; i \in [0, k] \tag{2.1}$$

where μ and σ are the mean and standard deviation of all rewards within the group. This formulation is well-suited for language models because generating k unique responses for a prompt is straightforward with modern decoding algorithms. Furthermore, it is more intuitive for prompt-completion settings than PPO's completion trajectories.

2.4.3 Direct Preference Optimization

Both PPO and GRPO require an explicit reward model standing between human preferences and the gradient signal, which adds a training stage and a source of approximation error. Direct Preference Optimization avoids this two-step pipeline by training directly on preference pairs.

This algorithm, DPO, is an alternative to RL-based alignment algorithms. Introduced by Rafailov et al. [2023], DPO aligns an LLM by directly optimizing its policy on pairs of preferred and dispreferred responses using a classification-style loss derived from the Bradley-Terry model. The loss implicitly recovers the same objective as RLHF but without training a separate reward model or running reinforcement learning. It is often found to yield outcomes comparable to or better than PPO at lower computational cost Xu et al. [2024]. DPO maximizes the following objective

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

where y_w is the preferred response, y_l the dispreferred one, π_{ref} the reference model (usually the SFT model), σ the sigmoid function, and β the temperature controlling how far π_θ is allowed to deviate from π_{ref}.

In summary, LLM alignment is a multi-step process that improves a model's task-specific capabilities and nudges it towards responses with a style and content that is preferred by humans.

2.5 Humor Generation with LLMs

With the theoretical and cognitive foundations in place, we now turn to the practical state of the art: how have large language models been applied to humor generation and understanding? The following reviews recent work on this question, with a focus on the persistent gap between what current models achieve and what humor

theories suggest is required. The success of (L)LMs over the past decade has also influenced the field of humor generation and understanding. In their recent survey, Song et al. [2025] frame humor understanding as an example for subjective and figurative language. They find that humor lags behind other areas, such as sentiment analysis, because it depends on incongruity, surprise, cultural knowledge, and non-literal meaning, which are much harder to infer reliably. Furthermore, the success of LLMs on certain humor benchmarks appears to be driven by generic joke patterns learned during training, according to the authors. LLMs can detect familiar jokes and *mimic* known humor styles, but not necessarily because they follow the mechanisms of humor theories faithfully. Song et al. call for multi-dimensional and theory-driven evaluation of generated humor, and call humor one of the strongest stress tests for genuine language understanding. Hessel et al. [2023] find that LLMs fall short of humans in ranking and explaining humorous captions from the New Yorker Cartoon Caption Contest (NYCCC). They conclude by advocating for modeling incongruity and cultural references in both humor understanding and evaluation. Kalloniatis and Adamidis [2024] conducted a systematic review on humor recognition. They find that the problem is among the most challenging in NLP, mainly due to the domain's subjectivity, which is amplified by noisy annotated datasets. Furthermore, the authors hypothesize that algorithmic improvements to LLMs will yield at most marginal gains on this problem. According to Kalloniatis and Adamidis, humor understanding should use preference pairs rather than binary labels or absolute scores. Lemmens and De Marez [2026] come to similar conclusions in their survey of state-of-the-art computational humor modeling. They stress the lack of integration of humor theory into LLM post-training steps, such as DPO or RLHF. Turgeman et al. [2025] tested whether LLMs are capable of *transfer learning* between different types of humor, including puns, dad jokes, and sarcastic headlines. They found that LLMs could, to some extent, transfer their humor-vs-non-humor discrimination ability to other types of humor. Importantly, their work suggests that training on heterogeneous datasets to include multiple types of humor improves generalization on held-out humor types. Recent work explicitly modeled humor theories in multimodal humor generation. Zhang et al. [2025] introduced HUMORCHAIN, a framework that performs multi-stage reasoning on humor theories such as incongruity detection, benign violation, and superiority to generate humorous image captions. Zhang et al. attribute their success to the explicit inclusion of humor theories that guide the model in selecting the core idea of the joke, and a trained humor discriminator model that is used iteratively to filter and improve captions. While many attempts on humor generation use decoder-only LLMs, Weller et al. [2020] trained an encoder-decoder model to translate news headlines in the Humicroedit dataset from 'normal' to 'humorous'. Their work demonstrates that humor preference datasets, such as Humicroedit, can also be used for humor generation. Weller et al. conclude that, although their approach was successful, automatic humor scores are unreliable and preference-based evaluation should be used. A similar result was found by Zhou et al. [2025], who trained a model to perform pairwise ranking of captions in the NYCCC dataset. Their results stress the importance of model alignment with human preference data, either via supervised fine-tuning (SFT) or DPO. The first attempt at using Humicroedit headlines and *generating* the single-word edit, rather than

predicting the edit's funniness, was done by Alnajjar and Hämäläinen [2021]. They operationalized humor as the coexistence of surprise and plausibility. Candidate replacement words were generated using a large syntactic repository of grammatical relations and evaluated for prosody, concreteness, surprise, and negativity. Generated headlines were found to be *slightly humorous* in only 36% of cases, and the authors highlight that better candidate-ranking mechanisms would improve results. Sakabe et al. [2025] pushed the frontier of humor evaluation by creating a six-dimensional humor annotation framework that includes novelty, clarity, relevance, intelligence, empathy, and overall funniness. They find that vanilla LLMs (not *specifically* trained for humor tasks) perform well below humans for generating creative or humorous text within their evaluation framework. Most importantly, they find that LLMs succeed at creating incongruities but fail to resolve them, due to a lack of empathy and social understanding. Work by Wang et al. [2024] proposes LoL (Leap-of-Logic), a two-stage training framework (SFT→DPO) that enhances humor generation by explicitly modeling multi-hop creative reasoning, combining instruction evolution, judgment learning, and preference optimization. Their ablations demonstrate that judgment training is crucial, that instruction evolution improves divergent thinking, and that DPO further amplifies gains through self-refinement. This work continued and improved the CLoT (Creative Leap-of-Thought, Zhong et al. [2024]) system. Kim and Chilton [2025] showed that equipping LLMs with explicit cognitive, social, and creative humor skills empowers them to generate meme captions rated on par with humans. They emphasize the effectiveness of first generating many diverse candidate jokes, which can be filtered and ranked.

Three threads run through this body of work. First, the surveys converge on a single diagnosis [Lemmens and De Marez, 2026, Kalloniatis and Adamidis, 2024, Song et al., 2025]: vanilla LLMs detect surface patterns of humor but do not faithfully implement the mechanisms named by humor theory, and prompting alone cannot close that gap. Second, the systems that do post-train (LoL, CLoT, HumorChain) all explicitly model incongruity, judgment, or leap-of-thought reasoning in the training signal. Thus, the operationalization of humor theory, not just better data, is what unlocks gains. Third, evaluation is the rate-limiting step: binary classification hides failure modes, scalar rewards are unreliable, and preference-based or theory-driven evaluation is repeatedly recommended. What is still missing across all of this work is a *formalism* that lets incongruity and resolution enter a reward function as continuous, differentiable signals. The next section introduces information theory, which provides exactly that.

2.6 Information Theory & Humor Quantification

Linguistic theories of humor describe concepts like incongruity, novelty, surprise, and relevance. Computational creativity research shows they matter empirically, but neither provides a formalism to operationalize them as numerical training signals for LLM alignment. Information theory offers exactly this: a framework for quantifying information. Because language encodes information through statistical dependencies, information theory, via measures such as mutual information, naturally lends itself

to analyzing concepts such as surprise and relevance in language.

Pointwise mutual information (PMI) measures the association between two outcomes, x and y. It quantifies how much more likely the pair (x, y) is to occur than you would expect if x and y were independent:

$$\mathrm{PMI}(x; y) = \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(y \mid x)}{p(y)}.$$

A PMI of zero represents independent outcomes; a negative value means the outcomes co-occur less often than if they were independent. Mutual information (MI) is the expectation of the PMI over the joint distribution of x and y. It can be interpreted as the average association of both outcomes, how much knowing one variable reduces the uncertainty over the other one:

$$I(X; Y) = \mathbb{E}_{p(x,y)}[\mathrm{PMI}(x; y)] = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)}.$$

PMI has a long history in NLP tasks: it has been used to locate co-occurrences in masking tasks Bouma [2009], quantify the cause of eye movements during reading tasks [Wilcox et al., 2024], and measuring topic coherence [Zheng et al., 2023].

As stated earlier, humor often involves creating incongruities. Past work has modeled this incongruency through the lens of information theory. Kao et al. [2016] built an information-theoretic computational model of pun comprehension using measures of ambiguity and distinctiveness. The results show that humor arises from distinct support for competing interpretations of a text, which can be quantified to accurately predict the text's funniness. Hwang et al. [2025] introduced BottleHumor, a system that uses the Information Bottleneck principle (simultaneously maximizing compression and relevance of information) to decide which knowledge fact is worth conditioning on when explaining the humor behind image-caption pairs. Hwang et al. find that repeated selection of information using their formalism improves humor understanding beyond chain-of-thought or self-refinement. Franceschelli and Musolesi [2025] further develops the idea of quantifying the creativity of LLM outputs using information theory. They introduce the Context-based score for Value and Originality (CoVO), which balances *value* (the response should be relevant to the input) and *originality* (the response should be unlikely/novel/surprising).

The CoVO score is defined as

$$s_{CoVO}(\mathbf{x}, \mathbf{y}, p) = \lambda_v \log p(\mathbf{x} \mid \mathbf{y}) - \lambda_o \log p(\mathbf{y} \mid \mathbf{x}).$$

Franceschelli and Musolesi showed that the CoVO score can be used as a numerical reward for RL methods by applying to three tasks

- Poem generation: given a tone and a style, generate a short poem. CoVO was used as the GRPO reward. The experiment showed that this fine-tuning step led to fewer repetitions of famous poem lines in the model's training data and improved the generated poem's adherence to metric constraints.

- Mathematical problem solving: the CoVO score was applied to the model's reasoning traces to reward diverse approaches, an external reward reinforced verifiably-correct answers. Fine-tuning with CoVO produced diverse reasoning traces without sacrificing accuracy, suggesting that the model learned different ways of reasoning towards the correct answer
- NoveltyBench: a common benchmark that tests a model's ability to generate diverse but high-quality responses to a prompt. Franceschelli and Musolesi found that CoVO added diversity to the prompt completions without sacrificing the quality of each individual answer.

These results show that CoVO can increase LLM output diversity *on tasks the model can already perform*. Our approach in this work is based on measuring surprise in language using PMI, as captured by the CoVO score. Although humor generation is thought to require creativity when performed by humans, it is a new application area that Franceschelli and Musolesi has not yet tested.

3

Methods

To answer our research questions, we propose a training pipeline with several components. The two main components are a learned humor judge $\mathcal{J}$ and a generator $\mathcal{G}$ that is supervised-fine-tuned and then post-trained with GRPO against a composite reward combining $\mathcal{J}$ with the information-theoretic CoVO score. We test this pipeline on two generative tasks: single-word headline editing (Humicroedit) and cartoon captioning (NYCCC). The previous chapter established the motivation for both components: a learned humor signal trained on human preferences rather than continuous scores, and an information-theoretic operationalization of incongruity-resolution that supplements that signal.

This chapter is organized in the order that the system was built rather than as a single static architecture, because the method evolved in response to empirical findings as the project progressed. We began with Humicroedit as a low-risk pilot, since its single-word edit constrains the output space and makes both $\mathcal{J}$ and $\mathcal{G}$ easier to analyze. However, early experiments exposed three limitations: a low humor ceiling (edits lean heavily on absurdity and niche political references), too few raters per headline for $\mathcal{J}$ to generalize reliably, and a small training corpus of fewer than 20,000 samples. We therefore shifted the main effort to NYCCC, whose larger and more diverse caption pool gives $\mathcal{J}$ a stronger training signal and $\mathcal{G}$ a richer output space. The same overall pipeline applies to both tracks; what differs is the input modality (text-only for Humicroedit, optionally vision-language for NYCCC), the CoVO reference distribution, and the auxiliary prompts that the value term uses to score topical relevance. On NYCCC we additionally explored a multimodal extension of $\mathcal{G}$ that used the cartoon image for caption generation, but it was ultimately set aside. After the GRPO arms turned out to be statistically indistinguishable from the SFT baseline, we ran a broad ablation sweep and a small human preference study to characterize the gap.

Section 3.1 defines the two main components in their dataset-agnostic form: the humor judge $\mathcal{J}$ and the SFT-then-GRPO generator $\mathcal{G}$, together with the CoVO reward that supplements $\mathcal{J}$ during GRPO. Sections 3.2.1 and 3.2.2 instantiate the pipeline on the two datasets, with NYCCC presented first as the main track and Humicroedit second as the secondary one. Section 3.3 then describes the two complementary evaluation modes: qualitative descriptive coding of generated outputs and a small human preference study.

3.1 Training pipeline

The training pipeline has two main components. A *humor judge* $\mathcal{J}$ serves as a reward model to rate the funniness of model outputs. The *generator* $\mathcal{G}$ produces humorous outputs for the given task. Figure 1 gives a top-to-bottom overview: human-rated examples train $\mathcal{J}$ and seed the supervised fine-tuning of $\mathcal{G}$; the resulting $\mathcal{G}_{\text{SFT}}$ is then frozen and serves both as the initialization of the GRPO policy $\mathcal{G}_\theta$ and as the reference $\mathcal{G}_{\text{ref}}$ for the CoVO reward, which combines with $\mathcal{J}$ into a single scalar that drives the GRPO update.

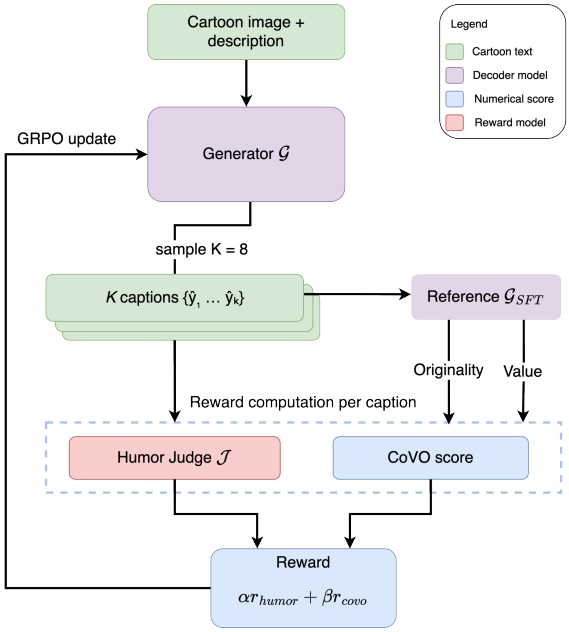


Figure 1: Schematic overview of the proposed training pipeline. The diagram reads top-to-bottom: human-rated examples feed both the humor judge $\mathcal{J}$ and the supervised fine-tuning of the generator $\mathcal{G}$; during the GRPO stage, candidate generations are scored by $\mathcal{J}$ together with the CoVO originality and value terms computed against a frozen reference model $\mathcal{G}_{\text{ref}}$.

3.1.1 Humor Evaluation

The proposed system generates humorous outputs that require automatic evaluation. We call the model responsible for this the *humor judge*, $\mathcal{J}$. In Figure 1, $\mathcal{J}$ is the left reward branch: it is trained once from human preference pairs and later supplies the humor reward to GRPO. Since standard regression on human-rated funniness scores is usually much harder than relative ranking of two outputs [Hossain et al., 2020a, Wang et al., 2024, Zhang et al., 2025], we train a Bradley-Terry (BT) model [Bradley and Terry, 1952]. For preference pairs $(y_i, y_j) \in \mathcal{D}^n$ where $y_i \succ y_j$, BT learns latent quality scores $\beta_i,\ i \in 1 \dots n$ for items $x_i,\ i \in 1 \dots n$ such that $\sigma(\beta_i - \beta_j) \to 1$, i.e. higher-preferred items receive higher scores. It does so by minimizing the BT cross-entropy loss

$$\mathcal{L} = -\sum_{(i,j)\in\mathcal{D}} \log \sigma(\beta_i - \beta_j).$$

A modern practice for BT models is to use neural networks to estimate the latent quality scores: $\beta_i = f(x_i)$ where f is a neural network. We train the BT model on preference pairs but use the latent quality scores as relative-ranking signals in the generator's reward. Thus, for an output y, $\mathcal{J}(y) = \beta_i$ gives a numeric score that can be used to rank different outputs relative to each other. The outputs for both tasks are represented as text (and image) tokens. Thus, we construct the neural network f with an encoder backbone and a regression head. All task-specific training details and parameter choices are explained in sections 3.2.1 and 3.2.2. We implement margin scaling in the BT objective. Margin scaling introduces a confidence weight m_{ij} derived from the score gap, such that the loss becomes

$$\mathcal{L} = -\sum_{(i,j)\in\mathcal{D}} m_{ij} \log \sigma(\beta_i - \beta_j).$$

Pairs where the human score difference is large receive higher m_{ij} weights, so they contribute more to the gradient and push the model to learn a larger $\beta_i - \beta_j$ separation for strongly preferred items. We furthermore add an anchoring term to the objective in the form of an MSE loss. This anchors β scores to the magnitude of human judgements, making the absolute output values more interpretable and comparable to other reward components. Thus, the complete loss is

$$\mathcal{L}_{\text{BT}} = \frac{1}{2}\sum_{i\in\mathcal{D}} (f(x_i) - s_i)^2 - \sum_{(i,j)\in\mathcal{D}} m_{ij} \log \sigma(f(x_i) - f(x_j)).$$

for human scores s, neural network f, and anchor weight $\frac{1}{2}$. We use this loss to train the humor judge $\mathcal{J}$ that assigns a score to each generated output.

3.1.2 Humor Generation

After defining the humor judge $\mathcal{J}$, we introduce the second component of the pipeline: the generator $\mathcal{G}$ whose outputs $\mathcal{J}$ scores during training. $\mathcal{G}$ occupies the central column of Figure 1: it is first supervised-fine-tuned on human-written examples to produce $\mathcal{G}_{\text{SFT}}$, then post-trained with GRPO against $\mathcal{J}$ and an information-theoretic auxiliary reward, while $\mathcal{J}$ itself is trained once on offline human ratings.

The main component of the pipeline is the generator model, $\mathcal{G}$, that generates humorous content in the task-specific format. The generator is based on a decoder-only pretrained language model. Specifically, we choose models from the Qwen3 family: Qwen3-4B-Instruct-2507 for text-only tasks and Qwen3-VL-2B [Team, 2025] for vision-language tasks. These instruction-tuned LLMs provide a good capacity-speed tradeoff. We adapt the pretrained models in two fine-tuning steps.

3.1.2.1 Supervised Fine-tuning

The first step is supervised fine-tuning to teach the generator the task structure. This method uses the train split of either dataset in a supervised fashion: the generator receives an input prompt x and predicts the output tokens y, while being updated via cross-entropy. The dataset inputs are embedded in task-specific instruction prompts

and documented in Appendix B. The main purpose of SFT is to give the generator a strong prior on the desired output format and humor style of each task. We use the HuggingFace prompt-completion chat template for both tasks [Face, 2024]. The templates ensure that the training loss is computed only on the completion tokens, not on the prompt.

3.1.2.2 Reinforcement Learning

In the second step, we optimize the model towards generating funnier outputs using reinforcement learning. We define a humor-specific reward function and use Group Relative Policy Optimization (GRPO, Shao et al. [2024]) to maximize the reward. GRPO is an RL algorithm that is commonly used for language models. For each input prompt x, $\mathcal{G}$ generates k completions $\{\hat{y}\}^k$. Each completion receives a reward score. GRPO then computes the advantage of each completion $\hat{y}$ relative to the others as (Eq. 2.1). For more details, read Shao et al. [2024].

The reward function combines a direct humor signal from the humor judge, $\hat{s} = \mathcal{J}(\hat{y})$, with the information-theoretic CoVO score. The next three paragraphs define each reward component in turn before discussing the optimization details.

The CoVO score. The CoVO term corresponds to the right-hand reward branch in Figure 1, which queries the frozen reference $\mathcal{G}_{\text{ref}}$ with the task prompt and an auxiliary inverse prompt to obtain the originality and value signals. Franceschelli and Musolesi [2025] define the CoVO score as

$$r_{\text{CoVO}}(x, y) = \lambda_{\text{value}} \log p_{\text{ref}}(x \mid y') - \lambda_{\text{orig}} \log p_{\text{ref}}(y \mid x). \tag{3.1}$$

where p_{ref} is the reference distribution, typically a frozen language model, and $\lambda_{\text{value}},\ \lambda_{\text{orig}} > 0$ are scaling factors. We choose the reference distribution to be a frozen copy of the post-SFT generator, $\mathcal{G}_{\text{ref}} := \mathcal{G}_{\text{SFT}}$. Two natural candidates exist for the reference distribution: the base model $\mathcal{G}$ (vanilla Qwen3-Instruct) and the post-SFT generator $\mathcal{G}_{\text{SFT}}$. We choose $\mathcal{G}_{\text{SFT}}$ because it provides a more demanding creativity signal. Since $\mathcal{G}_{\text{SFT}}$ has already been fine-tuned on humorous generations, its predictive distribution over tokens covers common humorous patterns. Measuring originality against this distribution pushes the GRPO-trained generator to produce tokens that are surprising even relative to a humor-aware model, rather than merely surprising relative to a general-purpose language model. Analogously, the value term asks whether $\mathcal{G}_{\text{SFT}}$ can recover the input to the task from the generated content; using a humor-tuned reference means that recovery must succeed despite $\mathcal{G}_{\text{SFT}}$'s bias toward funny completions, providing a stricter test of topical relevance.

In the original CoVO framework, x denotes the LLM input and y its output. For both tasks, x is a prompt that combines instructions with the data input. We explain what the terms represent in each task's context in subsequent sections. We now define the two reward components directly.

Originality. The originality reward measures how surprising the generated output

is under the reference model. We hypothesize that a low-probability replacement indicates more novelty or incongruency:

$$r_{\text{orig}} = \log \mathcal{G}_{\text{ref}}\Big(\hat{y} \mid x\Big).$$

Because this term enters the CoVO score with a negative sign (Eq. 3.1), a **lower** r_{orig} (i.e., a more surprising edit) yields a **higher** reward.

Value. The value reward measures how easily the input can be recovered from the generated tokens. We prompt $\mathcal{G}_{\text{ref}}$ with a task-specific auxiliary prompt, which instructs the model to do the task-specific inverse action. More details about this are in sections 3.2.1 and 3.2.2. The value reward is the log-probability that $\mathcal{G}_{\text{ref}}$ recovers the original input x:

$$r_{\text{value}} = \log \mathcal{G}_{\text{ref}}\Big(x \mid \hat{y}\Big).$$

A high r_{value} means the generation is topically anchored: despite being humorous, the input's topic is preserved.

The full CoVO reward is then:

$$r_{\text{CoVO}} = \lambda_{\text{value}} r_{\text{value}} - \lambda_{\text{orig}} r_{\text{orig}}. \tag{3.2}$$

As Franceschelli and Musolesi [2025] discuss in detail, the implementation of the vanilla CoVO score is problematic for autoregressive LLMs, because multiplying probabilities of individual tokens always biases shorter sequences. Hence, they propose a version that normalizes probabilities obtained from $p(x \mid y)$ and $p(y \mid x)$, considering the model's vocabulary V and the length of the input and output sequences.

Let $\hat{y} = (t_1, \ldots, t_L)$ denote the token decomposition of the humorous generation, and $x = (u_1, \ldots, u_M)$ the token decomposition of the input. We further define the prompt contexts

$$\begin{aligned} c_{\text{orig}} &:= \mathcal{P}_{\text{fun}}(x), \\ c_{\text{value}} &:= \mathcal{P}_{\text{reverse}}(\hat{y}) \end{aligned}$$

so that r_{orig} conditions on c_{orig} and r_{value} conditions on c_{value}. Because multiplying token-level probabilities in an autoregressive model biases shorter sequences, we adopt the length- and vocabulary-normalized variant proposed by Franceschelli and Musolesi [2025]:

$$r_{\text{orig}}^{\text{norm}} = \frac{1}{L}\sum_{i=1}^{L}\Big[\log \mathcal{G}_{\text{ref}}(t_i \mid c_{\text{orig}},\, t_{<i}) \;-\; \max_{v\in V} \log \mathcal{G}_{\text{ref}}(v \mid c_{\text{orig}},\, t_{<i})\Big], \tag{3.3}$$

$$r_{\text{value}}^{\text{norm}} = \frac{1}{M}\sum_{j=1}^{M}\Big[\log \mathcal{G}_{\text{ref}}(u_j \mid c_{\text{value}},\, u_{<j}) \;-\; \max_{v\in V} \log \mathcal{G}_{\text{ref}}(v \mid c_{\text{value}},\, u_{<j})\Big], \tag{3.4}$$

where V is the model's vocabulary. Each term inside the brackets measures how far the actual next-token log-probability falls below the best possible token at that position, averaged over the sequence length. This normalization makes the rewards comparable across generations of different token lengths.

The normalized CoVO reward is then

$$r_{\text{CoVO}}^{\text{norm}} = \lambda_{\text{value}}\, r_{\text{value}}^{\text{norm}} - \lambda_{\text{orig}}\, r_{\text{orig}}^{\text{norm}}. \tag{3.5}$$

Full reward. Using these definitions and the estimated humor score $\hat{s}$, the total reward is

$$r = \alpha\, \hat{s} + \beta\, r_{\text{CoVO}}^{\text{norm}} \tag{3.6}$$

where the coefficients λ_{value}, λ_{orig}, α, and β are weights, such that $\lambda_{\text{value}}, \lambda_{\text{orig}} > 0$, $0 < \alpha < 1$, and $\beta = 1 - \alpha$. Both summands of the reward are functions of the same generation $\hat{y}$, but measure different qualities of it.

The GRPO phase trains the generator to produce edits that maximize the predicted funniness and the originality of the generated output while staying relevant to its original topic. Franceschelli and Musolesi [2025] added a KL divergence term to the GRPO objective, penalizing divergence between policies, and found it to improve their results. We adopt this and compute the KL divergence between the trainable generator $\mathcal{G}_\theta$ and the frozen reference policy $\mathcal{G}_{\text{ref}}$. Algorithm 1 formally describes the GRPO training loop.

Reward magnitudes The terms in the composite $\mathcal{J}$ + CoVO reward have different magnitudes. $\mathcal{J}$ produces latent quality scores that, if anchored with an MSE loss, live on a similar scale as the human funniness scores. The CoVO term is a weighted sum of log probabilities, which can take a much greater range that may be difficult to estimate pre-training. In order to use the reward weights α and β effectively, both terms have to have a comparable magnitude. We use exponentially moving average (EMA) normalization with a momentum parameter of 0.1. By tracking the mean and variance of each reward component over a training batch, we z-score them to a mean of 0.5 and standard deviation of one. It is important to note that all validation and test-set rewards that we report are not normalized; they are computed after training is completed. The reason for this approach is to make sure that the rewards enter the model's update step with appropriate weights, but result metrics are undistorted and can potentially show the different growth curves of rewards.

Algorithm 1: GRPO Post-Training Loop

Input :
Dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$ of inputs (headlines or cartoons)
Generator $\mathcal{G}_\theta$ (trainable policy) and $\mathcal{G}_{\text{ref}}$ (frozen reference)
Humor judge $\mathcal{J}$
Hyperparams: $\alpha, \beta, \lambda_{\text{val}}, \lambda_{\text{orig}}$, group size K, learning rate η, KL weight γ
$\mathcal{P}_{\text{fun}}$, prompt to generate a funny output $\hat{y}$
$\mathcal{P}_{\text{serious}}$, reverse prompt to recover the input x from $\hat{y}$
Output : GRPO-trained generator $\mathcal{G}_\theta$

foreach $x \in \mathcal{D}$ **do**

`// Sample a group of candidate outputs from the trainable generator`

$\{\hat{y}^{(k)}\}_{k=1}^{K} \sim \mathcal{G}_\theta(\cdot \mid \mathcal{P}_{\text{fun}},\ x)$

for $k \leftarrow 1$ **to** K **do**

$r_{\text{humor}}^{(k)} \leftarrow \mathcal{J}(\hat{y}^{(k)})$

`// CoVO term`

$r_{\text{orig}} \leftarrow \log \mathcal{G}_{\text{ref}}(\hat{y}^{(k)} \mid \mathcal{P}_{\text{fun}},\ x)$

$r_{\text{value}} \leftarrow \log \mathcal{G}_{\text{ref}}(x \mid \mathcal{P}_{\text{serious}},\ \hat{y})$

$r_{\text{covo}}^{(k)} \leftarrow \lambda_{\text{val}} r_{\text{value}} - \lambda_{\text{orig}} r_{\text{orig}}$

$r^{(k)} \leftarrow \alpha\, r_{\text{humor}}^{(k)} + \beta\, r_{\text{covo}}^{(k)}$

`// Compute advantages within the group`

$\bar{r} \leftarrow \frac{1}{K} \sum_{k=1}^{K} r^{(k)}$

for $k \leftarrow 1$ **to** K **do**

$A^{(k)} \leftarrow \frac{r^{(k)} - \bar{r}}{\text{std}(r)}$

`// GRPO objective with KL regularization to frozen generator`

$\mathcal{L} \leftarrow -\frac{1}{K} \sum_{k=1}^{K} \Big(A^{(k)} \log \mathcal{G}_\theta(\hat{y}^{(k)} \mid \mathcal{P}_{\text{fun}},\ x) - \gamma\, \text{KL}\big(\mathcal{G}_\theta(\hat{y}^{(k)} \mid \mathcal{P}_{\text{fun}},\ x) \,||\, \mathcal{G}_{\text{ref}}(\hat{y}^{(k)} \mid \mathcal{P}_{\text{fun}},\ x)\big) \Big)$

$\mathcal{G}_\theta \leftarrow \mathcal{G}_\theta - \eta\, \nabla_{\mathcal{L}}$

Implementation details We used the following HuggingFace software packages to train all models: `Transformer Reinforcement Learning` (TRL, von Werra et al. [2020]) and `transformers` [Wolf et al., 2020], alongside `PyTorch` [Paszke et al., 2019]. Training was executed on high-performance clusters using Nvidia A40 GPUs.

3.2 Humor generation tasks

Past works in this field have used a variety of datasets and benchmarks to evaluate humor and creativity tasks. Most relevant for this study are Humicroedit, NYCCC, and Oogiri-GO. Humicroedit [Hossain et al., 2020a] was mostly used to evaluate humor ranking and understanding. Only a few (Alnajjar and Hämäläinen [2021], Weller et al. [2020]) generated humorous news headlines themselves. The NYCCC [Zhang et al., 2024] consists of more than 1.5 million humorous cartoon captions, crowdsourced from human submissions and paired with a continuous funniness score. It is commonly used for humor generation, understanding, and ranking. The Oogiri-GO dataset [Huang et al., 2025] is a large-scale multimodal and multilingual humor collection containing over 130,000 high-quality samples in English, Chinese, and Japanese derived from a traditional Japanese creative game. It consists of humorous text responses to inputs like images, text, or both, and features human preference annotations for approximately 78% of its entries. Previous work most commonly used it as a benchmark to evaluate and stimulate "Leap-of-Thought" (non-sequential creative reasoning) and humor generation capabilities in Multimodal Large Language Models.

We pursue two dataset tracks to validate our method. The main track uses the NYCCC dataset, the secondary one the Humicroedit dataset. Both frame the task of generating humorous content in a clear way and provide data samples with continuous, human-rated funniness scores. In the following subsections, we elaborate on the dataset-specific parts of our method.

3.2.1 The NYCCC track

3.2.1.1 Data

This dataset is split into training, validation, and test sets. Descriptive statistics are shown in Table 3.1. The complete information on the dataset can be found in Zhang et al. [2024]. A training tuple (x, y) consists of the cartoon image I, the cartoon description D including the cartoon location and involved entities, and a funniness score s.

3.2.1.2 Humor Evaluation on NYCCC

We use the vision-language model BLIP-2-OPT-2.7b [Li et al., 2023] as the backbone for the humor judge. Early experiments showed much stronger performance when the backbone utilizes visual and language signals from a cartoon contest. The regression head is a single output neuron with GELU activation [Hendrycks and Gimpel, 2016]. We freeze both the vision encoder and the language model of BLIP-2 during training.

	Train	Val	Test
Cartoons	140	44	47
Captions	874,357	267,563	298,770
Captions / contest (mean)	6,245.4	6,081.0	6,356.8
Funniness mean	1.214	1.212	1.217
Funniness std	0.126	0.123	0.125

Table 3.1: Descriptive statistics of the NYCCC dataset splits.

Not doing so leads to severe overfitting. As outlined before, our humor judge model is a Bradley-Terry model that trains on preference pairs, not individual samples. We construct the training preference pairs by first filtering out all captions with fewer than 50 votes. This ensures that humor scores come from a sufficiently large group of raters. We then group all captions by cartoon, and split them into the top 5% / middle / bottom 20% based on their funniness score. Next, we form candidate pairs by sampling either top×bottom, top×middle, or middle×bottom pairs uniformly at random with replacement. A candidate (*winner*, *loser*) is kept only if the funniness score gap exceeds our threshold t:

$$t \geq 1.5 \cdot \sqrt{\mathrm{SE}_w^2 + \mathrm{SE}_l^2}$$

where $\mathrm{SE}_w = {}^{\sigma_w}/_{\sqrt{n}}$ is the standard error of the winner caption (higher score) with n votes and a score standard distribution of σ. This threshold filters out pairs that are tied or very similar in score, and follows almost directly what Zhang et al. [2024] do. We use a lower threshold (1.5 instead of 3) to obtain more training samples. The sampling procedure is repeated until we obtain 100 unique pairs per cartoon contest.

The downstream purpose of this model is to reward generated captions. Within GRPO, this is essentially a ranking task: relative ordering of captions by their humor score matters most because GRPO normalizes rewards within a group. Thus, the most important metric during training of the humor judge is not MSE or even pairwise accuracy; it is Kendall's τ, a rank correlation metric on the range $[-1, 1]$. During a validation pass in training, we use the model to assign a latent quality score to each caption from the validation set. We then rank the validation set by its ground truth funniness scores, and again by the latent quality scores, and compute τ between these rankings. A value of 1 means the model's scores order captions exactly like the human ratings, 0 means the orderings are unrelated (random), and negative values mean the model orders captions opposite to humans. We compute the pairwise accuracy of the model as a secondary validation metric. The pairwise accuracy is simply the percentage of all validation samples where the model correctly assigned a higher latent quality score to the true winner caption. All relevant hyperparameters are listed in Section .

3.2.1.3 SFT and GRPO on NYCCC

Supervised fine-tuning on this dataset presents the following task: Given the text description of a cartoon, including its location, entities, and cartoon image, predict the

cartoon caption. We tested a multimodal decoder model, Qwen3-VL-2B, but found that it performed slightly worse in terms of validation perplexity and caption quality, compared to its text-only counterpart Qwen3-4B-Instruct. We thus proceeded with the text-only model. To train this model, we select the subset of training captions that are in the top 10% in terms of funniness score, resulting in a training set of 69581 captions. This step vastly improves training data quality by only training on the funniest captions. Early experiments validated that this leads to better results than including more training data at the expense of caption quality. We train the model for four epochs and LoRA adapters on all target modules. Captions are generated at a temperature of 0.7 and `top_p` = 0.9.

The GRPO policy is initialized from the SFT checkpoint. GRPO trains on the same top-10-percent training set. Evaluation during the training happens on two pools of validation samples: a random pool of five cartoons that is freshly shuffled every eval step; a fixed pool of 20 cartoons that is stable across runs and training steps. We don't evaluate on the full validation set because it would slow down training significantly. The random pool serves as an approximation of the performance on the full validation set. The fixed set of 20 captions can be used to observe how generated captions for the same cartoon change with more training steps.

Alternative preference optimization We use GRPO as the main RL algorithm to train the generator model, but we ablate this choice by testing Proximal Policy Optimization (PPO) and iterative Direct Preference Optimization (DPO).

The motivation for trying PPO [Schulman et al., 2017] is its explicit value function. GRPO does not need one because it computes advantages within a group of actions (responses), thereby removing observation (prompt) variance. In contrast, PPO samples trajectories consisting of actions taken after different observations; it uses a value function, typically a neural network, to estimate the value of a state, its future cumulative reward. All PPO-specific hyperparameters for our implementation are described in Appendix A.

Early results from the humor judge showed that placing a cartoon caption on a continuous humor scale based solely on human preferences is unreliable; however, classifying which of two captions is preferred showed moderate success. This led to the adoption of a Bradley-Terry model for humor evaluation, rather than a regression model. We applied the same idea to the generator model. Instead of trying to optimize a continuous reward signal using RL, we construct preference pairs of generated captions. A formal definition of the algorithm is . Our iterative version of DPO is a meta-loop of $K = 5$ rounds sitting on top of a frozen SFT reference $\mathcal{G}_{\text{SFT}}$: at the start of iteration k the current policy $\mathcal{G}_{k-1}$ samples $N = 50$ captions per cartoon, those captions are scored by the humor judge $\mathcal{J}$ and by the frozen SFT model acting as p_θ for CoVO, each component is z-normalized within cartoon, and a 50/50 humor + CoVO mixed-z ranking is built. Within each cartoon, we pair the top-12 captions with the bottom-12 by matched rank (rank-i chosen $\leftrightarrow$ rank-$(N-i+1)$ rejected), yielding 12 preference triples per contest for that iteration. A fresh LoRA adapter is then attached to $\mathcal{G}_{k-1}$ and trained for one epoch of DPO,

with the reference model pinned to the frozen SFT $\mathcal{G}_0$, rather than the previous iteration's policy. This is a load-bearing design choice, since using $\mathcal{G}_{k-1}$ as the reference would let the KL anchor drift with the policy. After training, the new DPO adapter is merged and then unloaded back into the base to produce $\mathcal{G}_k$, which serves as both the sampler and the LoRA host for iteration k+1. The CoVO reference stays pinned to SFT throughout, so the only thing that changes across iterations is the candidate distribution being preference-labeled — the reward signal itself is held constant. The reasoning behind this approach was to harvest the gains of preference-over-continuous-score optimization, over multiple rounds of caption-generation → preference selection → SFT.

We compare a total of seven generator models as the basis of our analyses. The models differ in their optimization target (SFT vs. DPO. vs. GRPO), and their reward constitution. Table 3.2 summarizes these generator arms.

3.2.1.4 CoVO for NYCCC

Using the CoVO score to score the outputs in this task requires careful definition. The autoregressive equations for the value and originality terms (Eq. 3.3 and 3.4) define a context c in which input x and output tokens y appear. For the NYCCC task, the input x is the instruction prompt plus the cartoon description, and the output y is the generated caption. The context of the originality term is the instruction prompt plus the cartoon description, $\mathcal{P}_{\text{fun-caption}}$, which is shown in Appendix B. Thus, the originality term asks, "How surprising is the caption given the cartoon description?" - low originality could be interpreted as a too-descriptive caption. The value term is trickier. It asks, "How relevant is the cartoon caption to the cartoon description?" - a low value can be interpreted as unrelated to the cartoon. As stated before in the CoVO paragraph, a language model is unlikely to generate the cartoon description and involved entities from its caption. We use an auxiliary prompt, $\mathcal{P}_{\text{reverse}}$, that instructs the model to describe a cartoon, its location, and entities, given the cartoon caption. All relevant prompts are shown in the Appendix B, and all relevant hyperparameters are listed in Section A.

CoVO for images The attentive reader may wonder how the CoVO score can be computed for vision-language models that work on image inputs. Since the value term tries to recover the input to $\mathcal{G}$ from its output, it has to have a distribution over image tokens that is conditioned on text. The VL model that we use, Qwen3-VL-2B, only has a distribution over text tokens. Thus, it could never recover the tokens of the cartoon image from the cartoon caption, no matter how "on-topic" it is.

In the following, we derive an information-theoretically equivalent formulation of CoVO's value term for VLMs.

In their derivation of CoVO, Franceschelli and Musolesi define pointwise mutual information (PMI) between input x and output y as

$$\text{PMI}(x, y) = \log p(x \mid y) - \log p(x) = \log p(y \mid x) - \log p(y).$$

Algorithm 2: Iterative DPO Refinement Loop

Input :
Dataset $\mathcal{D} = \{C_i\}_{i=1}^{N}$ of cartoons C
Generator $\mathcal{G}_0$ (SFT-initialized policy) and $\mathcal{G}_{\text{SFT}}$ (frozen reference)
Humor judge $\mathcal{J}$
Hyperparams: $\alpha, \beta, \lambda_{\text{value}}, \lambda_{\text{orig}}$, rounds $R = 5$, candidates $N = 50$, preference pairs $M = 12$
$\mathcal{P}_{\text{fun}}$, prompt to generate a funny caption
$\mathcal{P}_{\text{reverse}}$, prompt to generate a cartoon description from the caption
Output : Refined generator $\mathcal{G}_R$

for $k \leftarrow 1$ **to** R **do**
  $\mathcal{D}_{\text{DPO}} \leftarrow \emptyset$
  foreach $C \in \mathcal{D}$ **do**
    `// Sample candidate captions from current policy`
    $\{\hat{c}^{(n)}\}_{n=1}^{N} \sim \mathcal{G}_{k-1}(\cdot \mid C)$
    `// Score each candidate`
    for $n \leftarrow 1$ **to** N **do**
      $r_{\text{humor}}^{(n)} \leftarrow \mathcal{J}(\hat{c}^{(n)}, C)$
      `// CoVO term`
      $r_{\text{orig}}^{(n)} \leftarrow \log \mathcal{G}_{\text{SFT}}(\hat{c}^{(n)} \mid \mathcal{P}_{\text{fun}},\ C)$
      $r_{\text{value}}^{(n)} \leftarrow \log \mathcal{G}_{\text{SFT}}(\hat{c}^{(n)} \mid \mathcal{P}_{\text{serious}},\ C)$
      $r_{\text{covo}}^{(n)} \leftarrow \lambda_{\text{value}}\, r_{\text{value}}^{(n)} - \lambda_{\text{orig}}\, r_{\text{orig}}^{(n)}$
      $r^{(n)} \leftarrow \alpha\, r_{\text{humor}}^{(n)} + \beta\, r_{\text{covo}}^{(n)}$
    `// Z-normalize scores within cartoon, build preference pairs`
    $\tilde{r}^{(n)} \leftarrow \textsc{ZNorm}(\{r^{(n)}\}_{n=1}^{N})$
    for $m \leftarrow 1$ **to** M **do**
      $\mathcal{D}_{\text{DPO}} \leftarrow \mathcal{D}_{\text{DPO}} \cup \{(\hat{c}_{\text{chosen}}^{(m)},\ \hat{c}_{\text{rejected}}^{(N-m+1)})\}$
  `// Train one DPO epoch; reference is pinned to frozen SFT`
  $\mathcal{G}_k \leftarrow \textsc{TrainDPO}(\mathcal{G}_{k-1},\ \mathcal{D}_{\text{DPO}},\ \mathcal{G}_{\text{SFT}})$

Arm	**Algorithm**	λ_1	λ_2	α	β
$\mathcal{G}_{\text{SFT}}$	SFT	–	–	–	–
$\mathcal{G}_{\text{mixed}}$	GRPO	0.5	0.5	0.5	0.5
$\mathcal{G}_{\text{humor}}$	GRPO	0.0	0.0	1.0	0.0
$\mathcal{G}_{\text{CoVO}}$	GRPO	0.5	0.5	0.0	1.0
$\mathcal{G}_{\text{value}}$	GRPO	1.0	0.0	0.0	1.0
$\mathcal{G}_{\text{originality}}$	GRPO	0.0	1.0	0.0	1.0
$\mathcal{G}_{\text{DPO}}$	DPO	$0.5^\dagger$	$0.5^\dagger$	$0.5^\dagger$	$0.5^\dagger$

Table 3.2: Reward coefficients for each generator arm. $\lambda_1 = \lambda_{\text{value}}$ and $\lambda_2 = \lambda_{\text{orig}}$ weight the CoVO value and originality terms; α and β weight the humor and CoVO rewards in $r = \alpha\,\hat{s} + \beta\, r_{\text{CoVO}}$. $\mathcal{G}_{\text{SFT}}$ is the supervised initialization that all other arms start from and uses no reward, hence the dashes. †*dpo* is not trained with an additive reward; preference pairs are labeled by the same mixed humor+CoVO composite as the *mixed* arm, so the values shown are the ranking weights rather than additive reward coefficients. DPO's own temperature parameter ($\beta_{\text{DPO}} = 0.3$) is unrelated to the β column.

Since x, the input prompt, is fixed, we optimize for the y that maximizes mutual information:

$$y^* = \arg\max_y I(x, y) \tag{3.7}$$

$$= \arg\max_y \left[\log p(y \mid x) + \log p(x \mid y)\right] \tag{3.8}$$

where the value term, $\log p(x \mid y) = \text{PMI}(x, y) + \log p(x)$. $\log p(x)$ is independent of y, and can be dropped:

$$\arg\max_y \log p(x \mid y) = \arg\max_y \text{PMI}(x, y), \tag{3.9}$$

maximizing the value term is equivalent to maximizing PMI.

This leaves the value term as $v = \log p(x \mid y)$, i.e. it is equal to PMI when the input is constant. As stated above, if x is an image, $p(x|y)$ cannot be computed by a VLM that outputs only text tokens. Thus, we need another PMI-equivalent that operates on both images and text. We propose using the cosine similarity between image and text representations from a contrastive model such as CLIP [Radford et al., 2021]. The following derivation shows why this is theoretically justified.

CLIP is trained with a symmetric InfoNCE objective [Oord et al., 2018]. For a batch of N image-text pairs $\{(x_i, y_i)\}$, the image-to-text loss is:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_i \log \frac{f_\theta(x_i, y_i)}{\sum_j f_\theta(x_i, y_j)}$$

where f_θ is a learnable scoring function. Oord et al. show that the optimal scorer

satisfies

$$f^*(x,y) \propto \frac{p(y \mid x)}{p(y)} = \frac{p(x,y)}{p(x)p(y)} = \exp(\mathrm{PMI}(x,y)).$$

Oord et al. conclude that minimizing the InfoNCE objective maximizes a lower bound on mutual information (MI, the expectation of PMI):

$$I(X;Y) \geq \log N - \mathcal{L}^*_{\mathrm{InfoNCE}} \,.$$

In essence, the scoring function f recovers PMI up to a constant $C(x)$ after training

$$\log f^*_\theta(x,y) = \mathrm{PMI}(x,y) + C(x). \tag{3.10}$$

CLIP [Radford et al., 2021] parametrizes the scoring function as

$$f_\theta(x,y) = \exp\left(\frac{\cos(\varphi_I(x), \varphi_T(y))}{\tau}\right)$$

where φ_I and φ_T are the image and text encoders of CLIP and τ a learned temperature. The CLIP encoders produce L_2-normalized embeddings, so their inner product equals the cosine similarity. Combining CLIP's score function with Eq. 3.10 gives

$$\log\left(\exp\left(\frac{\cos(\varphi_I(x), \varphi_T(y))}{\tau}\right)\right) = \mathrm{PMI}(x,y) + C(x) \tag{3.11}$$

$$\frac{\cos(\varphi_I(x), \varphi_T(y))}{\tau} = \mathrm{PMI}(x,y) + C(x) \tag{3.12}$$

$$\cos(\varphi_I(x), \varphi_T(y)) = \tau\mathrm{PMI}(x,y) + \tau C(x). \tag{3.13}$$

$$\tag{3.14}$$

So CLIP cosine similarity, divided by temperature, estimates PMI up to an additive constant. Using this result in Eq. 3.9 gives

$$\arg\max_y \log p(x \mid y) = \arg\max_y \mathrm{PMI}(x,y) = \arg\max_y \cos\left(\varphi_I(x), \varphi_T(y)\right).$$

The terms $C(x)$, τ, and $\log p(x)$ are constant for the same x, and can thus be ignored when optimizing for y. This shows that optimizing the cosine similarity between text and image embeddings from a CLIP model is information-theoretically equivalent to maximizing the PMI of texts x and y.

3.2.2 The Humicroedit track

After describing how the pipeline works on the NYCCC cartoon captioning task, we now apply it to Humicroedit. The two tracks share the same components ($\mathcal{J}$, $\mathcal{G}_{\mathrm{SFT}}$, GRPO with $\mathcal{J}$ + CoVO) but differ in input modality, output granularity, and the form of the CoVO reference distribution; the points below highlight where the second track deviates from the first.

The Humicroedit dataset frames the humor generation as changing a single word in a serious news headline to create a funny headline. Early experiments on Humicroedit showed limited success. Qualitative inspection of the training data suggests the dataset has a low humor ceiling; edits rely heavily on absurdity and niche political knowledge, making the humor narrow in range. Furthermore, the small dataset size and low per-headline rater count limit the reliability of the humor judge as a reward signal in GRPO; ablation experiments on progressively larger training subsets showed steady performance increases, suggesting the regressor had not yet saturated. Consequently, the remainder of this study focuses on the NYCCC track. In the following, we outline the dataset-specific details of our method.

3.2.2.1 Data

The Humicroedit dataset contains roughly 25000 news headlines, edited words, and associated human-rated funniness scores. The dataset is provided in train, validation, and test splits such that all edits of an original headline belong to just one set, avoiding headline leakage. Following Hossain et al. [2020a], we fuse the Humicroedit dataset with the FunLines [Hossain et al., 2020b] dataset to create a larger training set. This auxiliary source contains similar data (news headlines, edits, and human-rated funniness scores). Dataset statistics are shown in Table 3.3.

	Train	**Val**	**Test**
Examples	17,747	2,419	3,024
Unique originals	13,921	1,668	2,066
Funniness mean	1.100	0.935	–
Funniness std	0.590	0.579	–

Table 3.3: Descriptive statistics of the Humicroedit dataset splits.

3.2.2.2 Humor Evaluation on Humicroedit

Analogous to the cartoon captioning task, we train a Bradley-Terry model for evaluating the funniness of edited news headlines. The training objective and setup are the same. Preference pair construction differs. On NYCCC, pairs were constructed from captions for the same cartoon, which was straightforward because every cartoon had thousands of captions. To maintain the within-headline construction on Humicroedit, pairs need to consist of two edits of the same original headline. Since each headline has an average of 1.1 edits, this results in fewer preference pairs. Hence, we do not filter pairs based on a minimum score gap, as this would reduce the training set size too much.

The backbone of the judge is the 139-million-parameter DeBERTa-v2-base encoder [He et al., 2020]; our results contain an ablation of backbones with different parameter counts.

In the background section on affective computing, we established that incongruities lead to humor only if they elicit emotion. We model this in our judge by handcrafting

emotion-related features. Following Annamoradnejad and Zoghi [2024], we use the NRC Emotion Lexicon [Mohammad and Turney, 2013] to compute emotion vectors for the original word in the headline w_z and its replacement $\hat{w}_z$. An emotion vector is a vector that contains a scalar value for each of the 10 emotions defined in the lexicon. Each scalar indicates how strongly the emotion is associated with the word. The emotion lexicon contains emotion vectors for a vast collection of English words. The two emotion vectors are concatenated with the headline encodings for the regression head, which features a single linear layer and GeLU activation. A detailed list of training parameters can be found in Appendix .

3.2.2.3 SFT and GRPO on Humicroedit

The main difference for this training step versus NYCCC is that Humicroedit requires single-word edits, whereas NYCCC is an open-ended generation problem. Furthermore, Humicroedit requires the system to choose the edit position z, the word in the headline that should be replaced. We follow Alnajjar and Hämäläinen [2021]'s approach in parts by making only the content words of the headline eligible for selection. Content words are here defined as all nouns, proper nouns, and verbs. SpaCy's English transformer-based part-of-speech tagger [Honnibal et al., 2020] is used to determine the part of speech of each word in the headline. After filtering for content words, we uniformly at random select the target word. The following example illustrates the approach:

Headline: "Trump fires FBI director before the election."

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Word	*Trump*	*fires*	*FBI*	*director*	*before*	*the*	*election*
POS	PROPN	VERB	NOUN	NOUN	PREP	ARTICLE	NOUN
Selectable	True	True	True	True	False	False	True

Table 3.4: POS-based target word selection example. Selectable tokens are colored in green, non-selectable tokens in gray.

The candidate pool contains all selectable tokens: $\mathcal{C} = \{\text{Trump, fires, FBI, director, election}\}$ and the target word is then drawn uniformly at random:

$$\texttt{target\_word} = \textsc{RandomChoice}(\mathcal{C})$$

During SFT, the edit word is not chosen but determined by the completion target. During GRPO, we use the method described above, choosing edit word w_z at position z in headline H. The replacement word $\hat{w}_z$ is generated using the generator model

$$\hat{w}_z \sim \mathcal{G}\left(\cdot \mid \mathcal{P}_{\text{fun}}(H, w_z)\right)$$

where $\mathcal{P}_{\text{fun}}(H, w_z)$ embeds the headline and replacement word in a prompt of task-specific instructions. All relevant prompts are shown in Appendix and .

3.2.2.4 CoVO for Humicroedit

As Humicroedit entails single-word edits, the interpretation of the CoVO terms is straightforward. The **originality** term is defined as

$$r_{\text{orig}} = \log \mathcal{G}_{\text{ref}}\Big(\hat{w}_z \mid \mathcal{P}_{\text{fun}}(H, w_z)\Big).$$

A low log probability of the replacement word in the context of the headline indicates high surprise. The **value** term measures how easily the original word can be recovered from the edited headline. We prompt $\mathcal{G}_{\text{ref}}$ with the auxiliary "seriousify" prompt $\mathcal{P}_{\text{serious}}(\hat{H}, \hat{w}_z)$, which instructs the model to replace $\hat{w}_z$ in $\hat{H}$ with a serious alternative. The value reward is the log-probability that $\mathcal{G}_{\text{ref}}$ generates the original word w_z:

$$r_{\text{value}} = \log \mathcal{G}_{\text{ref}}\Big(w_z \mid \mathcal{P}_{\text{serious}}(\hat{H}, \hat{w}_z)\Big).$$

A high r_{value} means the edit is topically anchored: despite the humorous substitution, the original word remains the most natural "serious" replacement, so the headline's topic is preserved.

3.3 Evaluation of the system

Evaluating subjective tasks like humor remains very challenging. Automated metrics rarely capture the complexity of the task [Zhang et al., 2025, Amin and Burghardt, 2020a]. We therefore evaluate our method in different ways.

The humor judge is validated on the test set by computing correlation and error metrics between $\mathcal{J}(\hat{y})$ (its rating of the generator's output) and the human funniness scores. Given the inherent subjectivity of humor, these values are interpreted as an upper bound on how reliably $\mathcal{J}$ can be a proxy for human judgments. Moreover, we analyze examples with the largest errors and the greatest disagreements with human ratings to identify the model's characteristic strengths and weaknesses.

We evaluate task-specific generations quantitatively and qualitatively. For the former, we utilize both $\mathcal{J}$ and CoVO as scoring mechanisms. These metrics are used to evaluate win rates between different configurations of generators and human-written captions. Outputs are analyzed qualitatively in great detail using descriptive coding techniques. The two subsections below describe these complementary evaluation modes: a structured qualitative coding scheme applied to NYCCC captions, and a small human user study used to cross-check the automated rankings against human preferences.

3.3.1 Qualitative Cartoon Caption Evaluation

To qualitatively characterize the generated cartoon captions, we follow a descriptive coding approach. Rather than sampling across the full test set, we analyze two test-set cartoons in depth (cartoons 604 and 621; see Figure 7). For each cartoon, we code every caption generated at the lower temperature setting of 0.8, pooled

across all seven generator conditions (*sft*, the five GRPO arms, and *dpo*). Since each condition contributes 100 captions per cartoon, this yields 700 coded captions per cartoon. We first read through all captions of a cartoon to identify recurring joke templates: groups of captions that rely on the same humor mechanism or framing. We then code each caption along four orthogonal dimensions, whose category sets were developed through an open-coding pass before full coding:

- Image grounding (ignores / shallow / integrated): whether the caption engages with the depicted scene. *Ignores* means the caption could apply to any cartoon; *shallow* references visible elements without engaging the scene's central tension; *integrated* engages the depicted setup substantively.
- Humor mechanism (none / observational / incongruity / pun-wordplay / superiority / benign violation): the dominant comic device, drawing on standard humor theories.
- Failure mode (none / generic / off-topic / nonsensical / malformed): if a caption fails, how it fails.
- Tone (NYer-like / colloquial / generic LLM): the surface voice of the caption.

We assign a single label per dimension to each caption. Where several humor-mechanism labels could apply, we resolve to the narrowest one using the precedence pun > superiority > benign violation > incongruity > observational > none. We report the results as code counts per cartoon, pooled across the seven generator conditions. Our approach has two main limitations. First, all captions were coded by a single coder, who is also the study's author. Second, we code only two cartoons, so the analysis should be interpreted as descriptive and suggestive rather than confirmatory.

3.3.2 Human user study

The qualitative coding above offers a fine-grained but single-coder view of what the system actually generates. We complement it with a small human user study, which lets us check whether $\mathcal{J}$'s automated rankings agree with human preferences.

As we alluded to in Chapters 1 and 2, the evaluation of generated humor is challenging and metrics can be unreliable. Therefore, we conduct a small-scale online study on the NYCCC dataset. The purpose of this study is to gather human preferences for cartoon captions that were generated by different methods. Specifically, we sampled 20 cartoons from the held-out test set and used the $\mathcal{G}_{\text{SFT}}$ and $\mathcal{G}_{\text{mixed}}$ models to generate captions for each cartoon. We added a third caption per cartoon by sampling from the top-10% of human-written captions for the specific cartoon. Participants were asked to rank the three captions by their perceived funniness. This setup allows us to compute win rates between the three generation sources: human vs. $\mathcal{G}_{\text{mixed}}$, human vs. $\mathcal{G}_{\text{SFT}}$, and $\mathcal{G}_{\text{SFT}}$ vs. $\mathcal{G}_{\text{mixed}}$, so we evaluate whether humans prefer captions from one source over the others. Figure 2 shows one of the 20 ranking questions. Ranking within groups of responses rather than rating them on a numerical scale is common

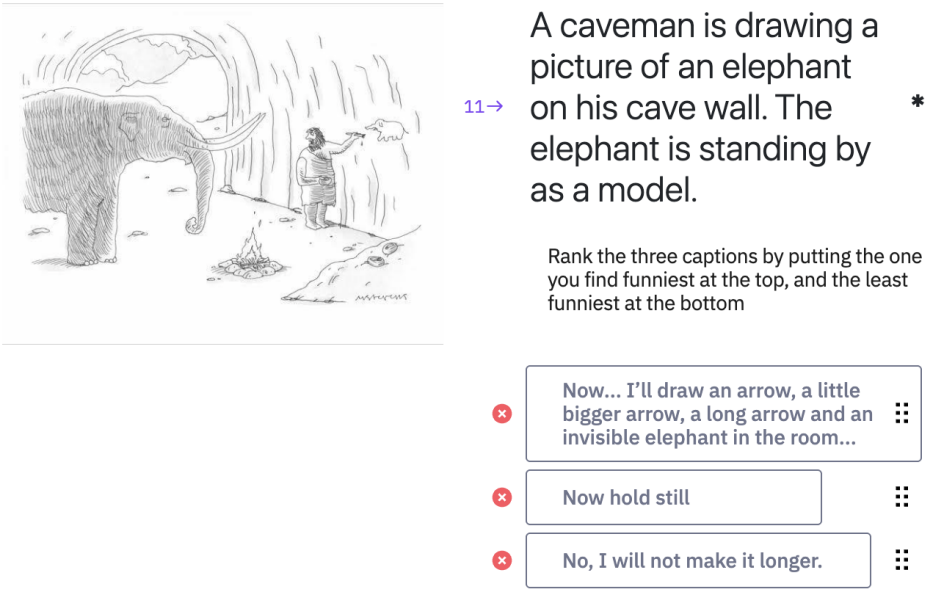


Figure 2: Example question from the user survey. Each of the 20 ranking tasks showed the cartoon image, the text description, the three generated or sampled captions in randomized order, and the instruction

practice in humor evaluation studies; humans are better at estimating funniness relative to other items. Chen et al. [2024], Zhang et al. [2025], Weller et al. [2020], Kim and Chilton [2025], Zhang et al. [2024] all utilize relative preference setups when conducting human user studies. Furthermore, the study features open-ended questions for the participants to describe their impression of the humor style of captions and how they think the humor could be improved. One of the questions asks participants whether each cartoon contained at least one caption that they perceived as funny.

We recruited a total of 77 participants, of whom we discarded 31 because they either did not give consent or did not respond to any of the 20 cartoon questions. The mean age of participants was 26.3 years. The detailed rating instructions for participants are in the Appendix C. We use the results of this study to check how human users' ratings align with our automated rankings via $\mathcal{J}$, and to perform qualitative analysis of the human feedback.

4

Results & Analysis

We present the results component-by-component rather than dataset-by-dataset, because conclusions about each component (humor judge $\mathcal{J}$, generator $\mathcal{G}$) transfer between the two tasks and because early findings on one component shaped later experiments on the other. The main NYCCC track is reported first: we evaluate $\mathcal{J}$ (Section 4.1, both quantitatively and qualitatively), then $\mathcal{G}$ (quantitative results, qualitative coding of generated captions, ablations that isolate the reward as the bottleneck, and a human-preference study). The Humicroedit track is reported last as a compact parallel analysis, focused on the points where its findings diverge from or reinforce the NYCCC story.

4.1 Humor judge $\mathcal{J}$

We analyze the humor judge $\mathcal{J}$ in two complementary ways. We first report ranking metrics across train, validation, and test splits, and break them down by funniness bin to characterize where on the funniness scale $\mathcal{J}$ is most reliable. We then inspect high-disagreement cases qualitatively.

4.1.1 Quantitative analysis

Since the main use case of the Bradley-Terry model $\mathcal{J}$ is ranking different outputs for GRPO, we evaluate it using pairwise-accuracy (PA) between two candidates, and Kendall's τ between sets of candidates ranked by their $\mathcal{J}$ score versus their ground truth scores. Table 4.1 shows the result statistics for each data split. Confidence intervals are computed using bootstrapping. Pairwise accuracy is computed within cartoons. This means that every (true winner, true loser) caption pair is drawn from the same cartoons. For Kendall's τ, we show both the pooled and the within-cartoon statistics. To compute the pooled version, all caption pairs are flattened and scored by $\mathcal{J}$ and their human score s; τ is computed across the whole array, so captions from different cartoons are compared. Downstream, $\mathcal{J}$'s scores are only used to rank captions *within* the same cartoon; we thus also report within-cartoon τ.

All metrics show a drop from train to validation to test set, with the biggest difference between train and validation. Notably, within-cartoon τ exhibits the smallest generalization gap.

Metric	Statistic	Train	Val	Test
PA	Mean ↑	0.6905	0.6552	0.6465
	CI low	0.6839	0.6432	0.6308
	CI high	0.6980	0.6670	0.6615
Kendall's τ_b (pooled)	Mean ↑	0.2816	0.2010	0.1779
	CI low	0.2637	0.1770	0.1493
	CI high	0.3011	0.2258	0.2074
Kendall's τ_b (per cartoon)	Mean ↑	0.1391	0.1313	0.1291
	CI low	0.1322	0.1230	0.1181
	CI high	0.1455	0.1394	0.1398

Table 4.1: Regressor performance across data splits. PA = pairwise accuracy (pair-weighted). Kendall's τ_b reported pooled across all pairs and averaged per cartoon. 95% bootstrap confidence intervals shown.

Figure 2 shows a scatter plot of predicted and true funniness scores. Because these scores live on different absolute scales (the predicted score is a latent quality score of a BT model), we z-score them to have a mean of zero and variance of one. This is done per-cartoon to account for inter-cartoon bias. The scattered points show roughly the same shape for all data splits: most data points form a cloud, which is most dense around the origin. This can be attributed to the z-scoring. Noticeably, the bottom-right quadrant shows a small vertical spread. Captions that humans scored as very funny (large positive true z-score) were rated within a narrow vertical range by $\mathcal{J}$; the model does not classify them as positive outliers. This is further supported by the vertical range of predicted scores: none of the data splits feature predictions greater than 5 on the z-score scale, whereas the human-rated predictions reach 10.

We further investigate the evaluator's performance on subsets of data binned by their ground truth funniness score. The score's range [1, 3] is binned into six intervals. To compute ranking metrics, we filter out bins with fewer than three captions. Figure 1 shows means and CIs for pairwise accuracy and within-cartoon Kendall's τ. In the left panel, we see that the average PA increases with the funniness bin on all training splits. The plot shows growing confidence intervals for funnier bins, which have much fewer samples than lower-funniness bins ($\sim$ 10000 samples in the lowest train-split bin vs. 1300 in the highest bin). PA on the test set is the only metric for which the mean decreases on the highest funniness bin. The right panel displays Kendall's τ, computed within-cartoon. On each split, the mean value decreases with increasing funniness bins. This metric also has wider confidence intervals for funnier bins with fewer samples.

4.1.2 Qualitative analysis

The aggregate metrics above show that $\mathcal{J}$ ranks captions above chance but struggles in the high-funniness tail. To understand what drives those errors, we now turn to a

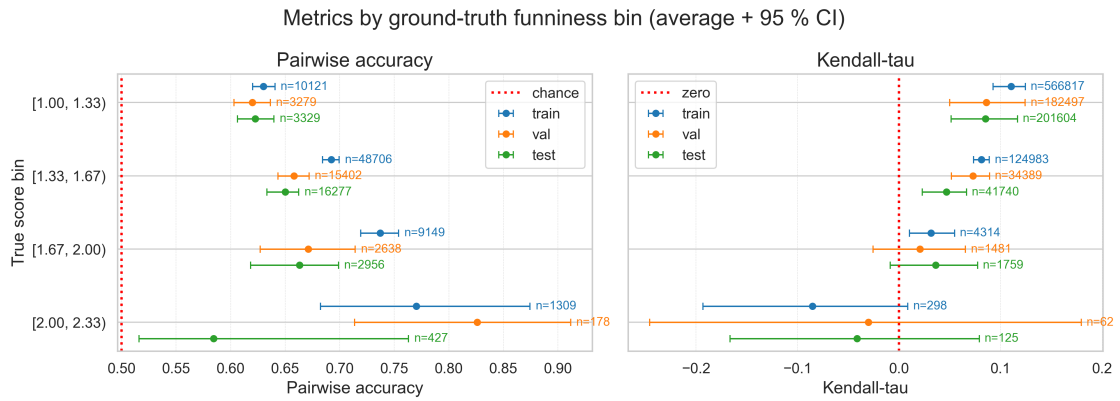


Figure 1: Metrics by ground-truth funniness bin. The plot shows average and 95% CIs. The red line indicates accuracy at chance or zero ranking correlation.

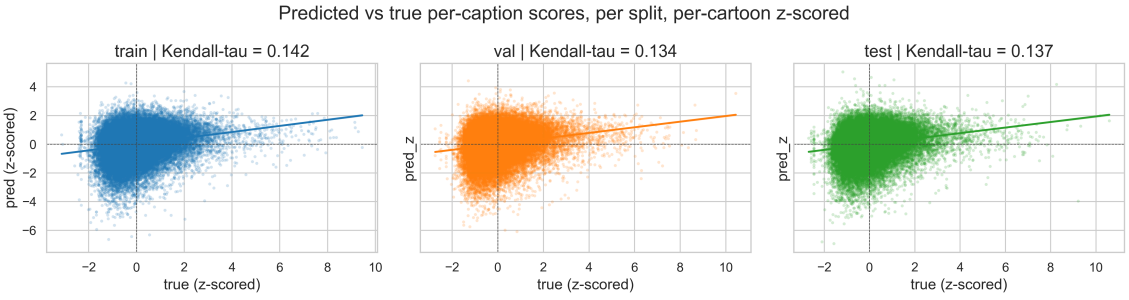


Figure 2: Scatter plot of predicted vs true funniness scores, z-scored within cartoons.

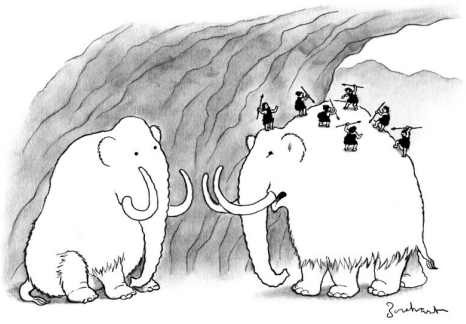


Figure 3: Contest #671. Human winner: *"Pachydermatologists"* ($\hat{y} = -0.86$). Human loser: *"I prefer it to the spray."* ($\hat{y} = +0.93$). The model prefers the generic dermatology joke; the winner requires recognising the hunters as skin parasites.

qualitative inspection of the cases where the model and human raters disagree most strongly, and of surface features that the model appears to rely on.

We started the qualitative analysis by looking at the winner-loser caption pairs for which humans and $\mathcal{J}$ disagreed most on, i.e. they predicted different winners with a large absolute margin: $|\Delta_h| + |\Delta_{\hat{y}}|$, one per cartoon, where the model ranked the human-preferred caption below the other (Table 4.2). Two patterns stand out. First, in five of ten cases, the model preferred a generic, format-conforming caption (short declarative sentence, no proper nouns, no typography) over a caption whose humor depends on a specific visual element of the cartoon (#600, #713, #671, #662, #747). In #671 (Figure 3), the human winner is a single large sentence that only lands once one sees the hunters camped on the mammoth's back; the model preferred the textbook dermatology joke "I prefer it to the spray." Second, the model penalizes captions that violate caption register: dialect spelling (#713), typos used as part of the joke (#579), atypical length (#677), and non-standard typography such as upside-down text (#747) all receive strongly negative raw scores. The register effect partly confounds the grounding one, since several visually grounded winners are also formally unusual. Three cases (#544, #699, #664) are not obviously model failures: both captions are image-relevant, and the disagreement reads as humor taste rather than a misread of the cartoon; the modest human margins (≤ 1.04) support this.

When we inspected the scored captions, we noticed that among those with very low scores, many contained the name or tag of the presumably caption author. We analyzed whether the presence of the author's name in the caption influences the score that it receives. We identified all captions containing the `@` character across splits ($n = 730$; 419 train, 166 test, 145 val) and re-scored each caption with $\mathcal{J}$ after stripping all text after `@` occurrences (usually the names). After dropping captions that became empty post-strip and deduplicating, $n = 642$ unique (split, contest, caption) triples were re-scored. Table 4.3 summarises the effect of stripping. Removing `@` shifts predicted scores upward by +0.38 on average, with 70.5% of captions receiving a higher score after stripping and only 25.8% a lower one. A Wilcoxon signed-rank test rejects the null hypothesis of no shift at $p \approx 2 \times 10^{-44}$. The magnitude of the shift (roughly $0.7\,\sigma$ relative to the prediction standard deviation

#	Human winner	$\hat{y}_w$	Human loser	$\hat{y}_l$	Δ_h	Pat.
600	Mission accomplished. We will report that this planet's inhabitants are friendly and playful.	+0.87	Just say you're from Norway.	+1.29	+3.00	G
713	. . . there's yer problem! ya gotta short circus!	−0.72	Goddamit Jenkins, quit clowning around and get back to accounting!	+0.23	+1.38	G,T
544	The New Yorker must be testing its new Caption Selection system.	−0.51	I'll have my guy email your guy.	+1.18	+0.61	?
671	Pachydermatologists.	−0.86	I prefer it to the spray.	+0.93	+0.39	G,T
579	he just got his resuslts from ancestry.com	−0.17	We switched to canned food.	+0.90	+1.05	T
699	Have you considered the possibility that you are three separate people?	−0.41	Okay, Id and Ego, how do you feel about what Superego just said?	+0.66	+1.04	?
662	Did you see the rock he gave her.	+0.44	Their religion precludes divorce.	+0.95	+1.57	G
677	Andrew Lloyd Webber, this is your conscience speaking, you should write a musical about cats . . . [∼145 ch.]	−0.73	We have to stop meeting like this.	+0.68	+0.64	T
747	Good morning to you too.	−0.31	That's the new guy Joe. He says it will take $1.9 Trillion to turn things around.	+0.63	+1.02	G,T
664	I'm the only one who can make the charges disappear, understand!?	+0.22	Again, that's not magic; that's obstruction.	+1.20	+0.96	?

Table 4.2: Ten largest model–human disagreements on the test split, one per cartoon. $\hat{y}_w, \hat{y}_l$ are raw BT scores for the human winner and loser (not comparable across cartoons; within each row what matters is the sign of $\hat{y}_w - \hat{y}_l$, which is negative for every row by construction). Δ_h is the human mean-grade margin (winner − loser). Pattern codes: **G** image-grounding failure, **T** tone penalty, **?** ambiguous (both captions image-relevant; humor taste).

of ≈ 0.55) indicates that the model has learned to systematically *penalise* captions containing `@`, a surface feature unrelated to humor.

Summary The humor judge achieves above-chance accuracy when predicting which of two captions for the same cartoon is funnier. It shows moderate ranking capabilities and difficulty in ranking captions that were rated as very funny by humans. Qualitative analysis revealed that the evaluator grounds itself in surface cues of the captions: conformity in terms of sentence structure and length, typos, and residual caption author tags.

4.2 Generator $\mathcal{G}$

Next, we present the results of the generator model $\mathcal{G}$. First, we evaluate the supervised fine-tuning stage. Then, we analyze the results of different GRPO

	Mean	Median	Std
Predictions with '@'	−0.283	−0.320	0.550
Stripped predictions	+0.099	+0.133	0.513
Δ (stripped − original)	+0.383	+0.393	0.583

Table 4.3: Effect of stripping @ characters on predicted scores ($n = 642$ unique captions). Wilcoxon signed-rank $p \approx 2{\times}10^{-44}$; 70.5% of captions score higher after stripping.

Figure 4: Training curves for SFT loss, entropy, and mean token accuracy.

configurations quantitatively and qualitatively. Lastly, we present the results of the human user study.

4.2.1 Fine-tuning the generator

As described in the Methods, the first step to training the generator is fine-tuning it on human-written cartoon captions to adapt it to caption-writing. Figure 4 shows training curves for the loss, entropy, and mean token accuracy. Loss and mean token accuracy show that the validation metric saturates before 10 training steps, while the training metric continues to improve. For the entropy, we see a saturation after 18 training steps, around the same value for training and validation. These curves indicated that the model learned the task format very quickly. Qualitative inspection of $\mathcal{G}_{\text{SFT}}$-generated captions confirmed that this checkpoint serves as a good warm-start for GRPO, the next step in the training pipeline.

4.2.2 Quantitative analysis of GRPO

We found that $\mathcal{G}_{\text{SFT}}$ serves as a stable warm-start. Next, we analyze whether the GRPO stage moves the generator in a meaningful direction. The following compares the seven post-GRPO arms against $\mathcal{G}_{\text{SFT}}$, scored by $\mathcal{J}$, CoVO, and their combination. We perform analyses at the level of aggregate scores and per-cartoon paired comparisons.

Next, we present the results of different generator configurations. We use each arm (see Table 3.2) to generate 100 captions for each test-set cartoon. This is done for two different generation temperatures, 0.8 and 1.2. We then score each caption using the humor judge $\mathcal{J}$, the CoVO reward, the originality term of CoVO, the value term of CoVO, and the linear combination of humor score and CoVO. The results,

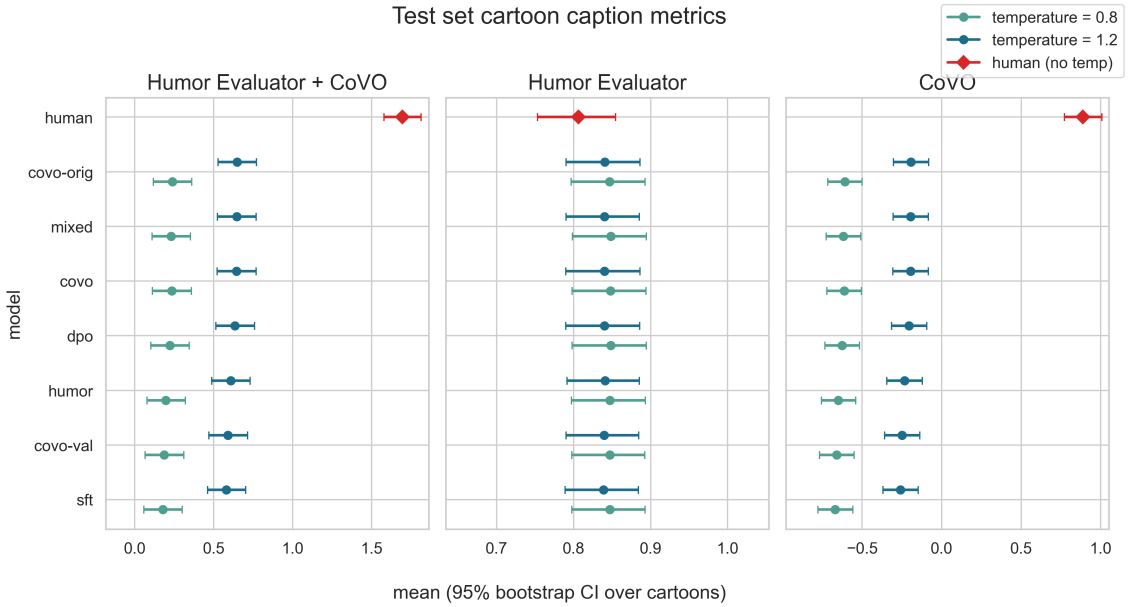
averaged over all captions and cartoons, are shown in Table 4.4. We also sampled 30 human-written captions per cartoon from the test set and score them in the same way. When scored by the humor judge, all arms show the same performance, within one temperature setting, means differ by a maximum of 0.02, which is one order of magnitude below one standard deviation of humor scores. Humor scores do not differ meaningfully between temperatures, but models reach a slightly higher mean at the lower temperature. For both temperatures, the originality-only arms reach the highest CoVO scores, the lowest (lower=better here) originality scores, and the best mixed scores. The best covo-value scores belong to the value-only arms. This serves as an important sanity check for GRPO mechanics: arms that were trained to optimize a singular metric (covo-value, covo-originality) score best in this metric. Figure 5 visualizes the numbers. The main panel 5a shows that all models have vastly overlapping CIs when scored by $\mathcal{J}$. Mixed scoring (equally weighted $\mathcal{J}$+CoVO) shows differences between arms, which are entirely caused by between-model differences in CoVO score. The second panel 5b reveals that CoVO-level differences in turn are caused by the originality term. The plot shows two trends: each model scores lower (=worse) at the lower temperature of 0.8 than at the higher temperature of 1.2. Secondly, there appear to be two groups of models: the arms $\mathcal{G}_{\text{SFT}}$, $\mathcal{G}_{\text{value}}$, and $\mathcal{G}_{\mathcal{J}}$ all have higher, so worse, originality terms than the $\mathcal{G}_{\text{originality}}$, $\mathcal{G}_{\text{mixed}}$, and $\mathcal{G}_{\text{CoVO}}$ models.

Temp.	Model	Humor ↑	CoVO Combined ↑	CoVO Value ↑	CoVO Originality ↓	Mixed ↑
—	human	0.806 ± 0.215	0.888 ± 1.161	-2.37 ± 0.419	-3.263 ± 1.112	1.695 ± 1.164
0.8	GRPO-COVO-ORIG	0.847 ± 0.199	**-0.609** ± 0.648	-2.368 ± 0.410	**-1.759** ± 0.529	**0.238** ± 0.687
	GRPO-COVO	0.848 ± 0.198	-0.613 ± 0.649	-2.367 ± 0.411	-1.754 ± 0.531	0.234 ± 0.691
	GRPO-MIXED	**0.848** ± 0.196	-0.618 ± 0.646	-2.367 ± 0.410	-1.749 ± 0.527	0.230 ± 0.684
	DPO	0.848 ± 0.199	-0.626 ± 0.646	-2.368 ± 0.412	-1.742 ± 0.526	0.222 ± 0.685
	GRPO-HUMOR	0.847 ± 0.199	-0.650 ± 0.630	-2.366 ± 0.411	-1.716 ± 0.509	0.197 ± 0.672
	GRPO-COVO-VAL	0.847 ± 0.198	-0.660 ± 0.637	**-2.365** ± 0.411	-1.704 ± 0.513	0.187 ± 0.678
	SFT	0.847 ± 0.197	-0.670 ± 0.629	-2.367 ± 0.411	-1.697 ± 0.503	0.178 ± 0.667
1.2	GRPO-COVO-ORIG	0.840 ± 0.198	**-0.193** ± 0.721	-2.371 ± 0.411	**-2.177** ± 0.613	**0.647** ± 0.755
	GRPO-MIXED	0.840 ± 0.200	-0.195 ± 0.732	-2.371 ± 0.413	-2.177 ± 0.627	0.645 ± 0.766
	GRPO-COVO	0.840 ± 0.198	-0.196 ± 0.729	-2.372 ± 0.412	-2.175 ± 0.619	0.644 ± 0.763
	DPO	0.840 ± 0.200	-0.206 ± 0.727	-2.372 ± 0.412	-2.166 ± 0.615	0.634 ± 0.760
	GRPO-HUMOR	**0.841** ± 0.197	-0.234 ± 0.713	-2.371 ± 0.413	-2.138 ± 0.599	0.607 ± 0.745
	GRPO-COVO-VAL	0.840 ± 0.198	-0.249 ± 0.711	**-2.369** ± 0.411	-2.119 ± 0.603	0.590 ± 0.747
	SFT	0.839 ± 0.200	-0.259 ± 0.701	-2.371 ± 0.411	-2.112 ± 0.588	0.579 ± 0.735

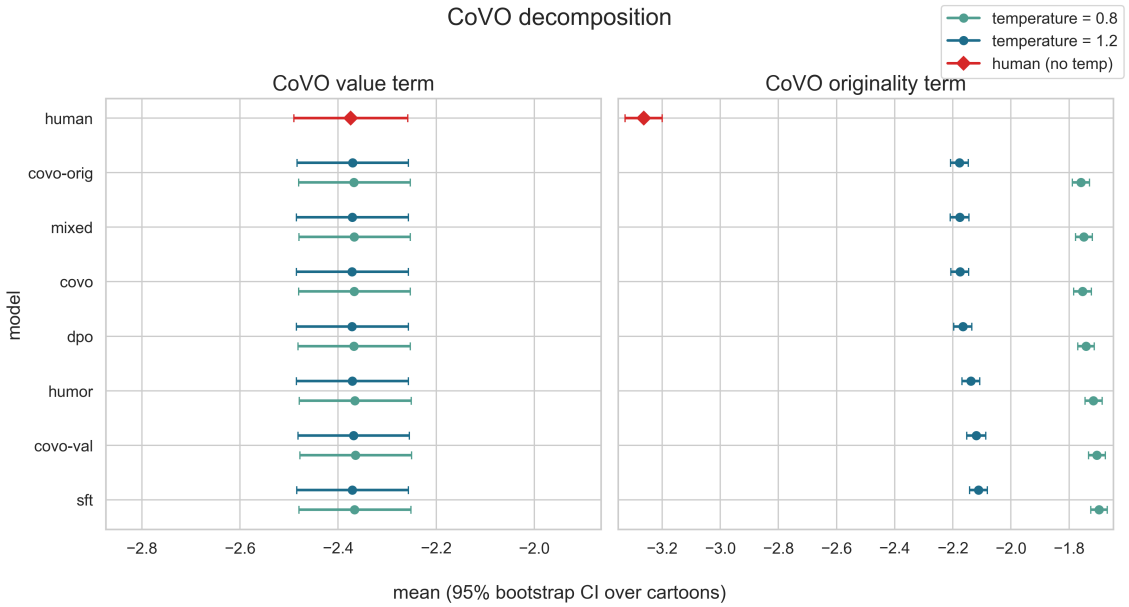
Table 4.4: Model evaluation results (mean ± std) across reward metrics at temperatures 0.8 and 1.2.

Table 4.5 shows win rates of different generator arms against human captions, rated by the judge $\mathcal{J}$. They result from averaging the winners of all 100 model-generated captions × 30 human-written captions per model arm. All win rates are > 0.5, indicating that, on average, all reward arms beat the human captions. This result has to be interpreted with great care. We found in the judge analysis that $\mathcal{J}$ relies on superficial features when ranking captions rather than deep humor quality.

To further analyze how the model arms differ, we compute pairwise comparisons per cartoon. For each pair of model arms, we count how many cartoons each model wins (mean score per cartoon is higher than the other), resulting in 47 cartoon-level



(a) Models scored by $\mathcal{J}$ + CoVO, $\mathcal{J}$, and CoVO



(b) Models scored by CoVO value and originality terms

Figure 5: Captions generated by all $\mathcal{G}$ models scored by different metrics. The panels show each model's mean score and 95% confidence interval.

Arm	T = 0.8	T = 1.2	Pooled
grpo-mixed	**0.6178**	0.5911	0.6045
dpo	**0.6178**	0.5918	**0.6048**
grpo-covo	0.6158	0.5924	0.6041
grpo-humor	0.6141	**0.5937**	0.6039
grpo-covo-orig	0.6137	0.5931	0.6034
grpo-covo-val	0.6136	0.5898	0.6017
sft	0.6135	0.5876	0.6005

Table 4.5: Win rate vs. human across training arms

paired comparisons per model pair. This is more honest than just comparing grand means because it accounts for cartoon difficulty as a paired factor. We use paired statistical tests (Wilcoxon signed-rank on per-cartoon means) and correct for multiple comparisons using the Holm-Bonferroni method. After correction, most pairwise comparisons on per-cartoon humor means do reach significance, but the absolute gaps are tiny (humor means range from 0.379 to 0.443 — a span of 0.064). The non-significant pairs cluster together: ‘covo‘ vs. ‘grpo-mixed‘, ‘covo‘ vs. ‘covo-orig‘, ‘covo-orig‘ vs. ‘grpo-mixed‘; i.e. the CoVO-weighted arms are statistically indistinguishable from each other. SFT and ‘covo-val‘ are reliably weaker on humor; ‘grpo-covo-orig‘ / ‘grpo-covo‘ / ‘grpo-mixed‘ form the top cluster.

Next, we analyze how the different reward dimensions correlate with each other. Therefore, we rank the models by each metric and compute Spearman correlations between the rankings. Figure 6 visualizes this as a heatmap. The strongest negative correlation (-0.71) is between the humor score $\mathcal{J}$ and the value score, indicating that model arms that rank high on one dimension do not do so on the other one. Value and originality have a moderate positive correlation of 0.54, and value and CoVO have a small correlation of 0.25, which is expected, since the value term is one of the two CoVO components. All other correlations are small.

Since the mean scores for the model arms are very similar, we suspect that the arms generated similar or identical captions. Upon testing, we found that the generated captions show significant overlap between model arms: $\mathcal{G}_{\text{CoVO}}$ and $\mathcal{G}_{\text{orig}}$ generated 75.4% identical captions (highest overlap), averaged across cartoons. The lowest overlap was between $\mathcal{G}_{\text{CoVO}}$ and $\mathcal{G}_{\text{SFT}}$ with 67.8% on average. This is a significant finding for how little GRPO moved the policies during training.

4.2.3 Qualitative analysis of GRPO

The quantitative results showed that the GRPO arms are statistically separable when scored by CoVO but indistinguishable from $\mathcal{G}_{\text{SFT}}$ on humor scores, and that the captions generated by different arms overlap heavily. To understand what the arms actually produce and why automated metrics struggle to separate them, we perform manual coding of caption content for two held-out cartoons.

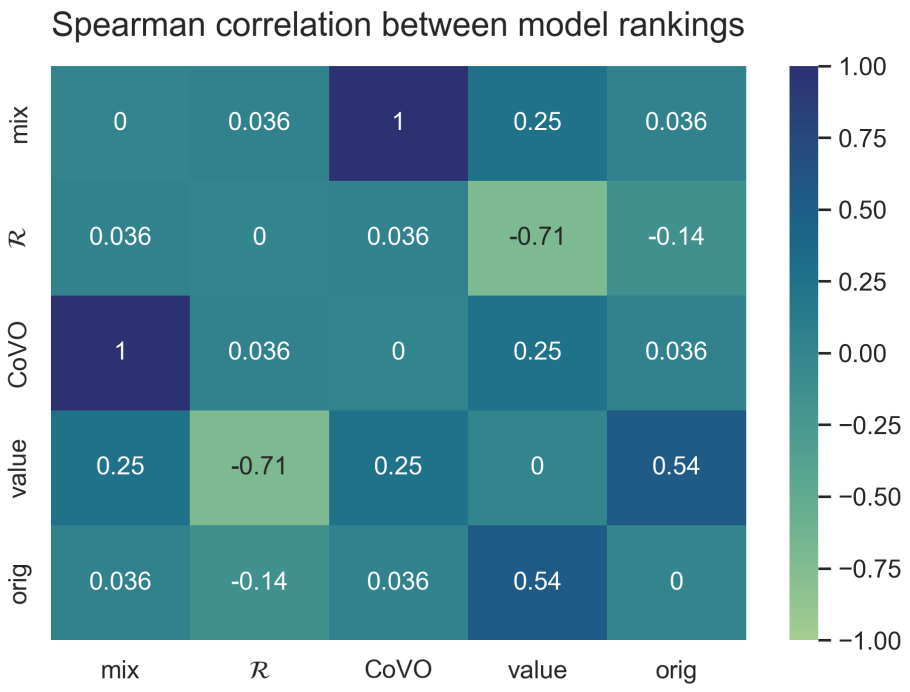
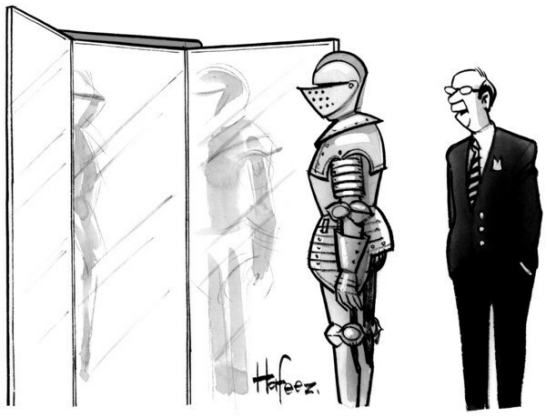


Figure 6: Spearman correlation of models being ranked by different scoring metrics

To analyze the caption-generation results qualitatively, we performed the answer coding described in Section 3.3.1 on cartoons 604 and 621. First, we went through each of the 700 captions per cartoon and identified joke templates: patterns that use the same humor mechanism. These are shown in Table 4.6. For the caveman-painting cartoon #604, the humor of many captions relied on framing the mammoth as a model and the caveman as an artist, or the "elephant in the room" idiom. For example, one of the captions reads "I'm just not a fan of the elephant in the room". The generator honed in on this idiom, but its application is still generic with respect to the cartoon; it relates to the mammoth, but not to the caveman painting it on the wall. For cartoon #621, most captions utilized the tailor's fitting commentary. The caption "Your helmet is perfect. I'm going to have to make you a new suit." suggests that the tailor is complimenting his client's look and is specific to the armor. However, the second sentence in the caption is unrelated to the armor; it is generic tailoring commentary, detracting from the humor. A joke pattern that we perceived as humorous was the "rusty pun" that allowed literal interpretations of the phrase "being rusty". Typical captions here read "I see you are a little rusty". While we found this creative, it relates to the knight's armor, but it is not tailor-specific. We found that, for both cartoons, while some joke templates were creative and humorous, they usually touched on just one aspect of the cartoon. This limits the humor potential, as highly rated human captions combine all the subtleties of a cartoon into a good joke.

Next, we analyzed the image grounding of captions. Each caption was classified into either ignoring the image (the caption could apply to any cartoon), showing shallow engagement (some cartoon elements are referenced), or deeply integrated

(a) **Cartoon #604**. Scene: A caveman is drawing a picture of an elephant on his cave wall. The elephant is standing by as a model. Location: a cave. Entities: Caveman, Mammoth, Cave painting.



(b) **Cartoon #621**. Scene: A knight is standing in front of some mirrors. A man in a suit stands behind him. Location: a tailor. Entities: Armour, Mirror.

Figure 7: Test-set cartoons selected for qualitative analysis. The captions are the descriptions from the dataset; this is all input that the generator receives for writing a caption.

image engagement (the cartoon's central tension is utilized). Both cartoons had the majority of captions in the shallow engagement category. For the mammoth-caveman cartoon, this typically meant relating to the painting activity or the mammoth. The caveman was seldom used, and fully integrated captions are rare. In the knight-tailor cartoon, integrated captions referenced both the knight's armor and the tension of a knight being in a tailor store. For example, the caption "I think you're taking the whole chivalry thing a bit too far." has the tailor advising the knight on fashion/image choices, which is stereotypical for tailors to do. Both cartoons featured many captions that were disjoint from the cartoon image. "And what if you're wrong", "I see you've been practicing" (both from the Knight-tailor cartoon), or "You think I'm an idiot?", "I just don't see how this is going to work" (mammoth-caveman) captions are all unrelated to the subtleties and central visual elements of the cartoons.

When analyzing the cartoon captions, we identified that they used different humor mechanisms. Table 4.8 lists them. *None* indicates that we classified a caption as entirely not humorous, which was the case for the majority of captions. The most common humor mechanism was general incongruity: the caption either emphasizes or resolves an incongruity that is set up by the cartoon scene. The mammoth-caveman caption "What part of 'draw me large' did you not understand?" relates the cartoon's small depiction of the mammoth on the cave wall to its large size. It further suggests that the caveman did not understand an instruction, relying on the common idea of cavemen being of inferior intelligence. The knight-tailor caption "I want to try on a different shade of steel" highlights the incongruence of trying on a knight's armor in a tailor store - a regular tailor can't customize armor.

Based on the findings above, it becomes obvious that many of the generated captions

Template	n	Example
Cartoon 604		
Size / proportions	60	What part of 'draw me large' did you not understand?
"Elephant in the room" idiom	46	I'm just not a fan of the elephant in the room.
Animal misidentification	67	I think I'm going to try the horse next.
Artist–model frame	69	This is going to be the last time I have a model.
First-time / amateur disclaimer	32	This is my first time drawing an elephant.
Anachronism	45	You must be the missing link.
Quoted-line / reported speech	14	...and then I said, "Can you be the model again?"
Unclassified	367	I don't know, I just like the way he's standing.
Cartoon 621		
Knight–corporate conflation	38	I don't see why I have to wear my helmet in a meeting.
Round table / King Arthur reference	44	I see you've chosen the round table.
Tailor's fitting commentary	128	I think you're taking the whole chivalry thing a bit too far.
Mirror / reflection wordplay	73	Your reflection wants to get dressed up.
"Knight in shining armor" idiom	36	Well, you're not a real knight in shining armor.
Rusty / steel / weapon puns	35	I see you're a little rusty.
Image-vs-substance reframe	38	You have the look, but you lack the armour.
Unclassified	308	And what if you're wrong?

Table 4.6: Caption templates for cartoons 604 and 621 (counts out of 700 generations per cartoon). Template values are cartoon-specific and therefore listed in separate blocks. Overlapping captions were resolved by specificity precedence within each cartoon.

are not humorous in any obvious way. We grouped captions into distinct failure modes: being too generic (not grounded in the cartoon, general); being too off-topic (unrelated to the image but not general); nonsensical (the caption contains nonsensical statements); and structurally or grammatically malformed captions (Table 4.9). For both cartoons, most failed captions were either too generic or off-topic. Both failure modes indicate that the policy's output does not use the cartoon scene as input. Some captions were generally nonsensical, for example, captions' second phrases directly contradicted the first: "No. Not 'Cave Elephant.' I said, 'Cave Elephant.'", "It's an elephant. It's not a mouse." (mammoth-caveman), or "Now you know why I'm a knight, not a knight." (knight-tailor).

Lastly, we analyzed the tone, or writing style, of captions. New Yorker captions have a distinct style: they are usually dry and written as a statement that is uttered by one of the cartoon characters. We found that for both cartoons, most captions did not match this style; they were generic, colloquial English prose.

Value	604	621	Example
Ignores image	162	179	I can't believe I'm seeing this! (#604)
Shallow engagement	386	338	I think I'll pass on the helmet. (#621)
Integrated with image	152	183	What part of 'draw me large' did you not understand? (#604)

Table 4.7: Image grounding codes for cartoons 604 and 621 (counts out of 700 generations per cartoon, pooled across the seven model conditions). *Ignores*: caption could apply to any cartoon. *Shallow*: references visible elements without engaging the scene's central tension. *Integrated*: caption engages the depicted setup substantively.

Value	604	621	Example
None	224	280	I'm not done.
Observational	192	162	I think he's just a little too big. (#604)
Incongruity	213	216	Your reflection wants to get dressed up. (#621)
Pun / wordplay	55	42	I'm not sure if you'll fit the bill. (#621)
Superiority	14	0	I'm going to make you look ridiculous. (#604)
Benign violation	2	0	I'll draw that after you eat his leg. (#604)

Table 4.8: Humor mechanism codes for cartoons 604 and 621 (counts out of 700 per cartoon). Single-label assignment with a precedence rule favouring narrower mechanisms (pun > superiority > benign violation > incongruity > observational > none) for captions where multiple labels could apply.

Summary The qualitative analysis of captions generated for held-out cartoons revealed several findings. GRPO arm policies overlap largely, illustrated by many of the captions being generated by several or all arms. The majority of potentially humorous captions relied on observational humor instead of using the tension of the cartoon scene to create incongruities. Pun and wordplay-driven jokes were often triggered directly by words ("elephant" in #604's caption triggered the idiom "elephant in the room"). Image-grounding of captions was mostly weak or absent, which rendered captions not humorous because they were either too generic or too close to a neutral statement about the cartoon.

All in all, we only identified a few captions that were humorous in the context of the cartoon and somewhat close to human-written ones.

4.2.4 Generator ablations

The qualitative coding revealed that GRPO training did not meaningfully change caption content across arms. The ablations below isolate why, asking systematically whether the optimizer, the reward model, or the reward design is responsible for the plateau.

Value	604	621	Example
None (no failure coded)	285	281	What part of 'draw me large' did you not understand? (#604)
Generic	184	162	I think I see a few things that could be improved. (#621)
Off-topic	121	148	I just wish we didn't have to work at the zoo. (#604)
Nonsensical	84	84	Nope. I'm still a mouse. (#604)
Malformed	26	25	Now you know why I'm a knight, not a knight.(#621)

Table 4.9: Failure mode codes for cartoons 604 and 621 (counts out of 700 per cartoon). *Generic*: caption could fit any cartoon contest. *Off-topic*: caption strays from the depicted scene. *Nonsensical*: caption is internally incoherent or does not follow from the image. *Malformed*: caption is grammatically or structurally broken.

Value	604	621	Example
NYer-like	69	81	It looks better in your cave. (#604)
Colloquial	469	468	I think I'll pass on the helmet. (#621)
Generic LLM	162	151	I'm sure you're going to look great in this outfit, but I think it'll be harder for the judges to see you in there. (#621)

Table 4.10: Tone codes for cartoons 604 and 621 (counts out of 700 per cartoon). *NYer-like*: dry, deadpan dialogue characteristic of New Yorker captions. *Colloquial*: natural conversational English without a distinctive voice. *Generic LLM*: long, over-explanation, or other surface markers of model-generated text.

The baseline generator for GRPO scored a mean humor score of $\mathcal{J}(\cdot) = 0.846$, essentially indistinguishable from the SFT initialization. $\mathcal{G}_{\text{mixed}}$ is configured as a text-only model (Qwen3-4B, LoRA $r = 32$), with the mixed humor $\mathcal{J}$ + CoVO ($\lambda_v = \lambda_o = 0.5$) reward, $\mathcal{G}_{\text{SFT}}$ as the reference model, $\text{lr} = 5 \times 10^{-7}$, $\beta = 0.04$, and a group size of 8. The ablations below systematically narrow down the cause. The single most important finding is negative, and explains why the trained reward model $\mathcal{J}$ prevents GRPO from improving. Although several of the ablations were not clean - they varied more than one variable at a time - they establish this finding from different angles.

GRPO mechanics If the GRPO implementation was incorrect, even a clean reward would fail to improve the policy. To rule out an implementation bug, we test a dummy reward

$$r = \begin{cases} 1, & \text{len(caption)} \geq 5 \\ 0, & \text{len(caption)} < 5 \end{cases}$$

in parallel to the true reward. The results showed that the toy reward rose by 26% over 500 steps, the KL increased from 0.11 to 0.32, indicating a learning policy. In

contrast, the humor reward arm stayed flat, and its KL barely changed, from 0.015 to 0.053 - a frozen policy. We checked if the algorithm's dynamics (advantages, clip fractions, gradients) are healthy. We found that advantages are healthy ($\mathcal{N}(0,1)$ shaped), and there were no groups with zero-variance advantages. Normalized gradients were stable around 0.8, and the clip fraction remained zero. The mixed training reward increased from 0.73 to 1.05 over the first 200 steps, then plateaued. We observed that the generator's vocabulary diversity, the diversity of words in generated captions, continued to decline from 0.17 to 0.108. This ablation showed that our GRPO implementation is functioning, but the humor reward doesn't produce a usable gradient under these settings.

KL penalty Having confirmed that the GRPO implementation itself can maximize a well-behaved reward, the first hypothesis for the frozen policy was that the KL penalty of $\beta = 0.04$ that Franceschelli and Musolesi [2025] used was extinguishing the reward gradient. We swept $\beta \in \{0,\ 0.005,\ 0.01\}$ on three reward arms (humor-only, covo-only, mixed) for 1500 training steps. Adapter weight movement was invariant to KL: the relative L_2 from SFT stayed in the 0.17—0.20% band across every β, including $\beta = 0$ (no anchor at all). Validation humor reward stayed at or just below the SFT baseline (0.846) in every cell, and the SFT tie-rate stayed pinned at 70—77% even with the KL penalty entirely removed. Removing the KL constraint did not free the policy. Removing the KL anchor did not free the policy, which meant the constraint had to lie in the optimization step itself. The most direct candidate was learning rate: if the reward gradient is real but the step size is too small to move the weights within 1500 steps, increasing LR should unlock policy movement.

Learning rate If the reward term in GRPO is not outweighed by the KL term, one reason for no policy change may be that the learning rate is simply too low. If this is the case, increasing the learning rate should unlock policy movement. We swept the learning rate across $\{5 \times 10^{-7},\ 5 \times 10^{-6},\ 5 \times 10^{-5},\ 5 \times 10^{-4}\}$ for humor-only, CoVO-only, and mixed reward functions. The results showed that raising the LR moved the weights monotonically — relative L_2 from SFT scaled from 0.18% at 5e-7 to 43—55% at 5e-4, and the mean humor rewards rose (humor-only hit 1.093 at 5e-5 vs SFT's 0.846) while tie-rate-vs-SFT dropped to 0%. Numerically, our hypothesis seemed to be confirmed. But inspecting the predictions revealed that the gains are pure reward hacking. The mode-collapse table is the decisive evidence: n_unique captions on the 44-cartoon validation set dropped from 44/44 at LR=5e-7 to $\frac{1}{44}$ at the higher LRs. Every collapsed cell converged to a single degenerate string. Examples of these degenerative captions are 'sorry' repeated 44 times (CoVO arms), 'myth myth myth ... ' token spam, whitespace-only, or 'Looks more like an miracle than an miracle'. The BT scorer rates these trivial strings *above* SFT's 44 distinct natural captions.

This highlights that the entire reward is highly fragile. The humor judge has degenerate maxima at trivial outputs; any GRPO configuration that can reach them collapses to them. At LR=5e-7, the policy stays near SFT only because the optimizer is too slow to reach those maxima within 1500-–3000 steps. This is essentially an

LR-induced regularization, not a property of the reward. A secondary finding: CoVO does not protect against collapse, it accelerates it (CoVO arms collapse at 5e-6, one order of magnitude earlier than humor_only at 5e-5), the opposite of the original "CoVO as a regularizer" hypothesis. The mixed humor+CoVO reward at high LR produced the worst cell of all (humor score 0.621, far below SFT even by the hackable scorer's own metric).

GRPO group size GRPO's group size parameter k controls the number of responses generated within a group. With KL and LR eliminated as the limiting factors at the stable LR=5e-7, the remaining candidate was the reward-gradient signal-to-noise ratio: larger groups yield cleaner per-step advantage estimates, which might unlock movement without requiring a destabilizing LR. We swept $k \in \{8, 16, 32\}$ with a humor-only reward for 1500 steps. The results showed that quadrupling the per-step advantage budget has no effect. The relative weight movement of the policy was 0.173%, 0.175%, 0.178%, respectively. The increase is just 0.005% relative to a fourfold change in group size. The validation humor score remained within 1 standard deviation of SFT, and captions remained unique.

The group size sweep is the third independent negative result on the same diagnosis, establishing a clear pattern: under the current reward model $\mathcal{J}$ (or CoVO), GRPO finds no usable gradients that do not lead to degenerate outputs, and increasing compute for larger groups does not help.

Multimodality After we experimented with GRPO's main tunable hyperparameters, we tested whether the architecture itself was the limitation, specifically whether a VL model as the generator that can condition on the cartoon image would break the plateau. We hypothesized that the text-only reward lacks a gradient signal because its captions are not grounded in the images, and $\mathcal{J}$ cannot assess them properly. First, we SFTed Qwen3-VL-2B to predict cartoon captions from their cartoon images. $\mathcal{G}_{\text{SFT-VL}}$ served as the initialization and reference model for running GRPO with $\mathcal{G}_{\text{VL}}$ a CoVO reward. The CoVO value term used the version derived in paragraph 3.2.1.4. The $\mathcal{G}_{\text{SFT-VL}}$ reached a validation loss of 2.97, 8% higher than the text-only SFT checkpoint, and a validation token accuracy of 45%, slightly below the baseline's 47%. Thus, the image modality contributes negligibly to base SFT quality. We then used the VL SFT checkpoint to run GRPO with the CoVO-VL reward. The VL results show that even with image-grounded rewards, the plateau persists, which is consistent with the main claim that the reward is the ceiling: the VL reward is still computed over the same BT humor scorer whose degenerate maxima exist in both text and VL form.

Table 4.11 shows that the training reward during GRPO decreases instead of increasing. Furthermore, the policy drifts toward the base reference, indicated by the rise in the originality term. The policy's collapse is further evidenced by the diversity ratio during validation, which measures the fraction of unique captions generated for a given cartoon. The low ratio shows that the policy converged on a small set of high-probability tokens, and thus generates the same captions over and over. On the positive side, we see a slight increase in humor reward on validation cartoons.

Signal	Start	End	Δ
training reward	18.6	15.7	−18%
originality term reward, $\log p$ under SFT-VL	−10.86	−8.51	+2.4
Humor ($\mathcal{J}$, not in loss)	—	—	+0.07
Training-side entropy	7.26	7.45	+0.19
Eval-time `diversity_unique_ratio`	0.76	0.32	collapse

Table 4.11: Training dynamics of GRPO-VL

This could be because the vision-integrated CoVO reward grounds captions better in the corresponding images. However, a manual inspection of generated caption samples revealed that this is likely not the case. The captions referenced objects in the cartoon images but seemed more descriptive than humorous.

This ablation showed that GRPO degrades its own training reward and drifts toward the reference even when the reward is vision-grounded, and the policy is multimodal. Image-conditioning the generator does not change the outcome, ruling out a text-only generator architecture as the cause of the plateau.

CoVO reference model In the Methods, we describe and justify why we chose $\mathcal{G}_{\text{SFT}}$ as the reference model for CoVO. We tested this choice by running GRPO with the vanilla Qwen3 model as a reference. This experiment did not result in significant policy changes; it hit the same humor validation mean, and adapter weight movement was minimal.

SFT data quantity We tested the hypothesis that a smaller, higher-signal subset (lower top-percent of captions per contest) would improve the SFT validation perplexity. Since the training set for NYCCC is large, we could filter for even higher-rated captions to train on without compromising size. Our results showed that more data is monotonically better (eval loss 10% > 5% > 2% > 1% of the data), with every condition peaking at roughly one effective epoch through unique data. This is a clean negative result for the "quality over quantity" hypothesis and a positive recommendation to keep top_percent=0.10 (or test higher). It does not bear on the GRPO plateau, but is included because it is a pre-registered hypothesis that failed.

Reward-model retraining Disabling early stopping and training the humor judge for the full 10 epochs produced a new best ($\tau = 0.201$, pairwise accuracy = 0.654), a +0.010 τ gain over the prior SOTA that had been cut off prematurely at epoch 5. This was used as the canonical downstream reward model.

CoVO-only training Training with humor fully zeroed (value-only, orig-only, balanced) produced policies, $\mathcal{J}$ rates above the human caption baseline and on par with a humor-trained policy, with no collapse to scene-paraphrase (value-only) or nonsense (orig-only). This supports the thesis claim that CoVO's value/originality balance is largely aligned with the BT humor signal rather than orthogonal, with

the important caveat that CoVO is computed over an LLM that was SFT-tuned on funny captions, so it is implicitly humor-correlated through the initialization. (Single-seed; the 0.005 cross-arm spread is within the GRPO seed-noise band, so the three-way ordering needs ≥2 seeds before any strong claim.)

4.2.4.1 Reward-shaping ablations

We found that the humor judge creates a noise floor that hinders the policy from learning. Thus, we ran some reward-related ablations to see if the floor can be broken.

Margin-gated advantages Evaluating the reward model $\mathcal{J}$ showed that it achieves much higher accuracy on caption pairs with large margins. We hypothesized that zeroing within-group advantages whose z-scored distance from the group mean is below a threshold δ_z should let the policy ignore the ranking where $\mathcal{J}$'s accuracy is close to chance, resulting in a better gradient signal. To test this, we swept $\delta_z \in \{0.0,\ 0.5,\ 1.0\}$. Results showed that no gated arm beat the vanilla control on the humor reward. Thus, per-rollout advantage gating does not extract additional signal from the $\mathcal{J}$ floor.

Anti-hacking augmentation The LR sweep showed that higher learning rates promote reward hacking of the humor judge and result in degenerate captions. We hypothesized that $\mathcal{J}$ is susceptible to these hacks because its training data only included well-formed captions. We augmented the humor judge training set with degenerate captions: empty strings, single-token repeats, punctuation clutter, and token loops, all associated with a humor score of zero. We pre-committed a sanity-check: the BT model trained on the augmented dataset has to score real captions higher than the degenerate ones in at least 80% of all validation samples. We aborted the GRPO stage of this experiment after two ratios of true-to-augmented sample ratios, 1:1 and 2:1, both failed to clear the 80% barrier; captions like "sorry" ×44 still outscored natural captions. During the training, the model learned to distinguish synthetic from real captions near perfectly (0.999 accuracy), but accuracy on human-written captions dropped from 0.814 to 0.624, because the grade floor squashed all top-tier human captions into a tight band. Reward model augmentation with synthetic degenerate captions cannot close the degenerate-maxima problem at the augmentation settings that we tested.

CoVO reward contribution Prior experiments showed that none of the reward arms (humor-only, mixed, CoVO-only) could meaningfully move the policy. To examine the statistical robustness of this, we performed a sweep across random seeds for caption generation ad training. Specifically, we trained five reward arms (humor-only, covo-only, mixed, covo-originality, and covo-value) on three different seeds each, and evaluated captions generated under three different random seeds, resulting in a 5 arms × 3 train seeds × 3 gen seeds = 45 setup. We found that the maximum pairwise $|\mu_i - \mu_j|$ across the five arms was 0.0053, smaller than the mean across-arm σ of 0.0103; every arm sat within $\pm 1\sigma$ of the SFT mean (0.846). 71-–74% of (cartoon,

gen-seed) pairs were literally identical captions to SFT, with a median margin of exactly zero for every arm. No arms suffered from mode collapse. Two subtleties: bootstrapped CIs showed the two single-term CoVO arms (value-only, orig-only) were marginally worse than SFT (CI excluded zero), while combining them in covo-full or adding humor (mixed) erased that small penalty; and weight movement saturated by step 250. At this configuration, no reward variant is distinguishable from SFT or from any other variant. The correct framing is not that the CoVO reward is intrinsically weak, but that at lr=5e-7 nothing escapes SFT, and at any working LR everything reward-hacks; the CoVO contribution question remains unanswerable until the reward is anti-hacked or retrained.

4.2.4.2 Learning algorithm ablations

If GRPO were specifically the problem, a different algorithm could move the generator model away from SFT. We tested PPO and DPO, and neither of them did, which supports the diagnosis that the reward, not the optimizer, is the ceiling.

PPO We hypothesized that PPO's learned value baseline might extract more signal from the noisy humor+CoVO composite than GRPO's group-relative baseline, breaking the plateau. We ran standard PPO in the same setup as the other ablations. We found that its held-out trajectory evaluation already peaked at step 50 (5% of the run), with humor at 0.8775 (+1.9% over baseline), and then drifted back below baseline by step 350. This is the same peak-then-regress shape as vanilla GRPO, with an even narrower productive window. On the post-hoc $\mathcal{J}$ evaluation, PPO was tied on the test split (+0.0015) and slightly worse on the validation pooled mean (-0.0054) and macro-mean (-0.0066). We conclude that PPO does not beat GRPO on this reward. This follows the common ablation thread: the value head does not make the humor judge's flat-gradient region traversable.

DPO From our results, we conclude that single-shot DPO is the one bright spot: $\beta = 0.3$ improved test mixed_z by +0.051 over SFT (-0.286 $\rightarrow$ -0.235), with no length blowup and perfect diversity. But the improvement is almost entirely due to CoVO, not humor (humor: 0.845 $\rightarrow$ 0.851, essentially flat because SFT was trained on top-rated captions and is near-saturated). The iterative loop is a clean negative result: iterations 2–3 regress back to SFT (-0.275 to -0.280), undoing the iter-1 gain, and this is robust across $t \in \{0.7,\ 1.0\} \times \{3,\ 5\}$ iter. The build-time captions reveal the hack. At temperature 1.0, the policy drifts into verbose English (2.5-times-longer captions that make CoVO explode and humor crater); at temperature 0.7, it drifts into a multilingual token soup. Yet, at the production evaluation temperature (0.8), the outputs look completely coherent, so the pathology is invisible on the evaluation distribution. The within-cartoon z-normalized composite reward has no "coherent English" floor: any drift direction that scores well on z(humor)+z(covo), including out-of-distribution token sequences, gets reinforced. Evaluation-temperature sampling tends to wash out these directions, but the policy is still displaced from the iter-1 sweet spot. This is the canonical failure mode for self-rewarding loops on a synthetic information-theoretic reward, and it independently confirms (from the

preference-optimization side) that the reward, not the algorithm, is the problem.

Summary Across KL, learning rate, group size, advantage gating, scorer augmentation, reward-component isolation, and two alternative RL/preference algorithms, the result is the same: under the canonical humor judge, no variation drives a defensible improvement over the SFT-initialized reference. The optimizer is healthy, the KL anchor does not prevent movement, and per-step compute is irrelevant; what fails is the reward gradient, which is alive for 200 steps and then either exhausts (at a safe LR) or leads straight into the scorer's degenerate maxima (at any working LR). The collapses are trivial strings that the scorer rates above genuine captions; scorer-side augmentation cannot remove them without destroying the natural-caption discrimination, and both PPO and iterative DPO reproduce the plateau through their own pathologies. Our claim is therefore not that CoVO is decorative, but even stronger: the $\mathcal{J}$ scorer cannot be used as a GRPO reward without anti-hacking measures, because the whole reward is fragile.

4.3 Human User Study

As outlined in the methods, we performed a user study to determine how humans rank captions by their perceived funniness. We collected a total of 77 responses, of which 31 were discarded because they did not meet the inclusion criteria. The mean participant age was 26.3 years, with a minimum of 17 and a maximum of 59 years, and 85% of participants were younger than 30 years. Each participant ranked three captions for each of the 20 cartoons, resulting in $46 * 20 = 920$ ranking responses.

Table 4.12 shows the win rates of each caption source. Human-written captions (taken from the NYCCC test set) won two-thirds of all cartoons. SFT-sourced captions were scored slightly higher than GRPO-sourced captions.

Caption Source	**Win (%)↑**	**Loss (%)↓**	**Mean Rank ↓**
Human	66.4	12.6	1.46
SFT	18.9	41.9	2.23
GRPO	14.7	45.4	2.31

Table 4.12: Win/loss rates and mean rank across 920 human preference judgements. Lower mean rank is better.

We use Kendall's W, a coefficient of concordance, as the metric for inter-rater agreement. A score of zero represents no agreement, and one represents total agreement. We report the winner agreement as a secondary metric. This measures the percentage of rater pairs that agree on the winner (i.e. rank the same caption highest). Figure 8 shows the distribution of W and the winner agreement across the 20 cartoons. We see that W spans a large range, with some cartoons having near-zero (random) agreement, and some reaching high agreement rates above 0.6.

To analyze what distinguishes captions that humans find funny from ones that they

Ranking Order	Frequency (%)
Human > GRPO > SFT	33.5
Human > SFT > GRPO	32.9
SFT > Human > GRPO	12.5
GRPO > Human > SFT	8.5
SFT > GRPO > Human	6.4
GRPO > SFT > Human	6.2

Table 4.13: Distribution of full preference orderings across 920 judgements (> denotes preferred over). Bold font indicates the best mean score in one metric over all models.

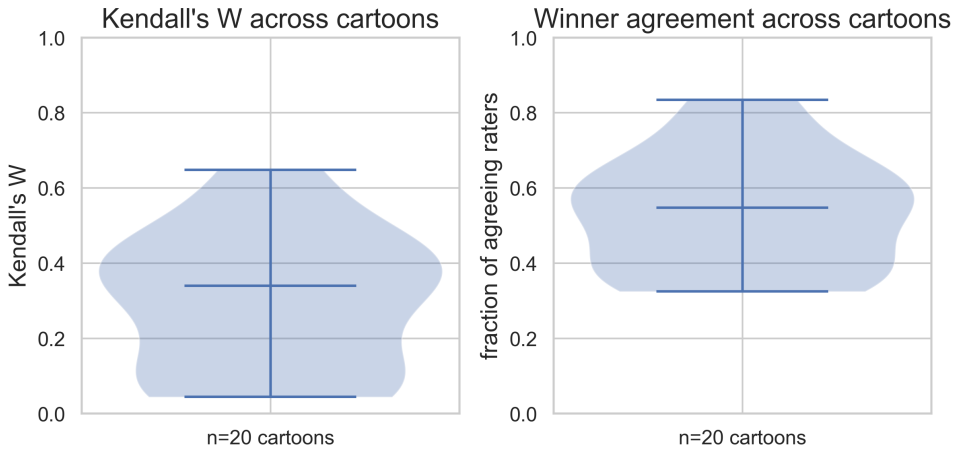


Figure 8: Distribution of Kendall's W inter-rater agreement between cartoons.

do not, we test the influence of syntactical features. We found that caption length does not play a role; high-W and low-W groups have almost identical caption length (54 vs 59 characters).

We analyzed the four cartoons, which showed very low inter-rater agreement ($W < 0.1$). In cartoon #548 (Figure 9), the human caption strongly relies on the contextual knowledge that salt lakes are primary testing grounds for very fast cars or vehicles with rocket engines due to their flat surfaces. Both generated captions use a generic route-deviation setup for their humor mechanisms, which requires no contextual knowledge and is not very rocket-specific. For this cartoon, the human caption is objectively better because it engages with the cartoon on a deeper level of interpretation. If human raters are unaware of the salt lake-rocket connection, the caption may seem as generic as the other two to them. We believe the unequal distribution of important contextual knowledge is the primary driver of low inter-rater agreement in the cartoons.

Table 4.14 shows the caption win results grouped into high and low inter-rater agreement groups. Human beats SFT in 74.9% of judgments overall ($p << 0.01$), rising to 83.1% on high-W cartoons and falling — but still significant — to 59.6% on low-W ones. Human beats GRPO even more decisively: 78.9% overall ($p << 0.01$),

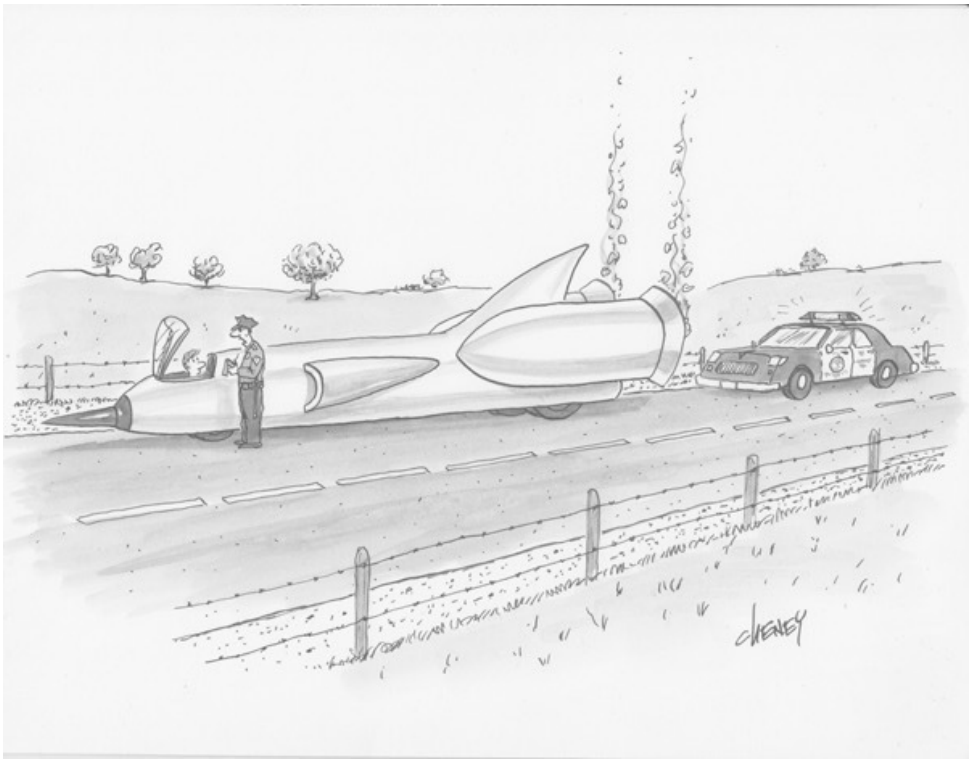


Figure 9: Cartoon #548
Human caption: The turnoff to the Salt Lake is in Utah. This is Kansas.
SFT caption: The GPS said this route would save me a month of gas
GRPO caption: The GPS said this route was safe, so I figured I'd take a crack at it.

Subset	Comparison	Win Rate (%)	Sign Test p
All ($n = 920$)	Human > SFT	74.9	1.48×10^{-53}
	Human > GRPO	78.9	6.42×10^{-73}
	SFT > GRPO	51.8	2.77×10^{-1}
High ($W > 0.24$, $n = 598$)	Human > SFT	83.1	9.19×10^{-64}
	Human > GRPO	87.0	4.00×10^{-81}
	SFT > GRPO	55.0	1.58×10^{-2}
Low ($W \leq 0.24$, $n = 322$)	Human > SFT	59.6	6.53×10^{-4}
	Human > GRPO	64.0	6.01×10^{-7}
	SFT > GRPO	46.0	1.64×10^{-1}

Table 4.14: Pairwise sign test results across all judgements and stratified by inter-annotator agreement (W). Win rate counts how often the left-hand system was preferred; p-values are two-tailed.

87.0% on high-W, 64.0% on low-W. SFT vs GRPO is essentially a coin flip. Overall SFT > GRPO at 51.8% (p=0.28, not significant). It nudges to 55.0% (p=0.016) on high-W cartoons and to 46.0% (not significant) on low-W ones. The GRPO stage did not produce captions that humans prefer to the SFT baseline — if anything, the high-W slice slightly favours SFT. The Human gap shrinks substantially on low-W cartoons (60-–64% vs 83—87% on high-W), which fits the reading that "low agreement" cartoons are ones where the human caption itself is mediocre, leaving more room for the models to compete.

After ranking all cartoon captions, participants were asked whether they found at least one caption per cartoon funny. Out of 42 participants that completed the question, 26 (62%) answered no, and 16 (38%) said yes. This is a strong indicator of how funny participants found New Yorker cartoon humor, as each cartoon had a human-written headline.

Ten participants responded to the open question that asked them how they would describe the humor type. Three participants mentioned dry humor, two described it as pun-driven. Other mentioned adjectives were "absurd", "cultural reference-based", "creative", "not dark", and "not present". Next, participants were asked in an open-ended question what would improve the humor quality for them.

Open-ended responses were coded thematically using a predefined codebook developed inductively from the response set. Two passes were made: first to define topic categories, then to assign each response to its best-fitting category. Responses that did not clearly fit any category were retained as unclassified. The results are shown in Table 4.15. We show a representative response for each category. These qualitative results reveal several interesting findings. Some participants found captions too literal and descriptive, while others were missing more obvious connections between the caption and the cartoon. While this seems contradictory at first, these impressions do not have to come from the same cartoon. The answers also underline the subjectivity

of humor. One participant wished for less dry humor; another for more.

Theme	***n***	**Representative response**
Caption length	4	*"Some of them were too long, making them lack the delivery. Some of them were just too complex; they could be simplified without losing the edge."*
Caption surprise	3	*"Some more unexpected outcome"*
Relatedness to image	5	*"Be more related to the picture. I often got the feeling it missed some details in the drawing that made the image absurd."*
Pop culture / modern references	4	*"It would be interesting to have more cultural/pop references to make the humor more relatable and less dry."*
Descriptive captions	5	*"Some captions were just basic descriptions, no pun or funny lines."*
Darker humor	4	*"Adding more dark humour."*
Wordplay	3	*"Word jokes / ambiguity."*
Other	6	"More dry humor"
Total	**34**	

Table 4.15: Thematic coding of open-ended survey responses to the question "I think the funniness of the captions could be improved by ...". Responses were coded manually against a predefined codebook. n indicates the number of responses assigned to each theme. Responses not fitting any theme are listed as 'other'.

Summary The user study revealed through 920 ranking responses that human-written captions are greatly preferred over generated ones. Furthermore, $\mathcal{G}_{\text{SFT}}$-generated ones were ranked slightly higher than captions generated by $\mathcal{G}_{\text{GRPO}}$. Free-text feedback from users revealed that the humor overall was often not their preferred style, and even human captions were not regarded as funny. Users named potential improvements by making the captions more image-grounded, including cultural references, and shorter.

4.4 Humicroedit Results & Analysis

As stated in the methods, most efforts focused on the NYCCC data set track. The following presentation and discussion of the results on the Humicroedit dataset is therefore a minimal version of what was done for NYCCC.

4.4.1 Humor judge $\mathcal{J}$

We analyze the BT-style humor evaluation model with the same methodology as the NYCCC track. Since Humicroedit's test set split is used for competitions, it lacks a funniness-score annotation and is therefore excluded from this analysis. Table 4.16 shows pairwise accuracy and Kendall's τ on train and validation splits. Both metrics show severe overfitting - train set performance is strong; the evaluator is nearly perfect at predicting which of two edited headlines is preferred over the other one. It can also be used to rank edited headlines with high correlation to human rankings. The performance on the validation set degrades markedly, with PA dropping to barely above chance and ranking correlations near zero. The overfitting gap is further supported by the scatter plot showing z-scored funniness score predictions, Figure 10. The left panel shows good calibration on the train set, indicated by a nearly linear slope. The validation line in the right panel, however, has a small slope. This shows that validation set predictions correlate poorly with the ground truth. The forest plot in Figure 11 confirms that the overfitting gap is present in all ground-truth funniness bins: train metrics are much higher than validation metrics. Furthermore, there is no apparent trend that the model performs better for any funniness bin. Only the bin's $[1, 1.5)$ PA confidence interval clearly excludes 0.5 (chance level), and for none of the bins τ CIs exclude 0 (no correlation).

Metric	**Statistic**	**Train**	**Val**
PA	Mean ↑	0.973	0.568
	CI low	0.967	0.535
	CI high	0.977	0.601
Kendall's τ_b (pooled)	Mean ↑	0.720	0.07
	CI low	0.715	0.038
	CI high	0.726	0.106
Kendall's τ_b (per cartoon)	Mean ↑	0.924	0.141
	CI low	0.910	0.07
	CI high	0.937	0.217

Table 4.16: Regressor performance across data splits. PA = pairwise accuracy (pair-weighted). Kendall's τ_b is reported pooled across all pairs and averaged per cartoon. 95% bootstrap confidence intervals shown.

4.4.1.1 Ablations on the humor judge

In the process of improving the judge, we experimented with several interventions.

Backbone capacity We started with a DeBERTa-v2-base encoder, a 139-million-parameter model. We hypothesized that the poor performance in early experiments was caused by the model's limited expressive power. Thus, we switched to the 900-million-parameter DeBERTa-v2-large encoder. This caused severe overfitting, indicated by a drop in validation accuracy. This led us to continue with DeBERTa-v2-base and vary the number of backbone layers that we wanted to freeze. Frozen

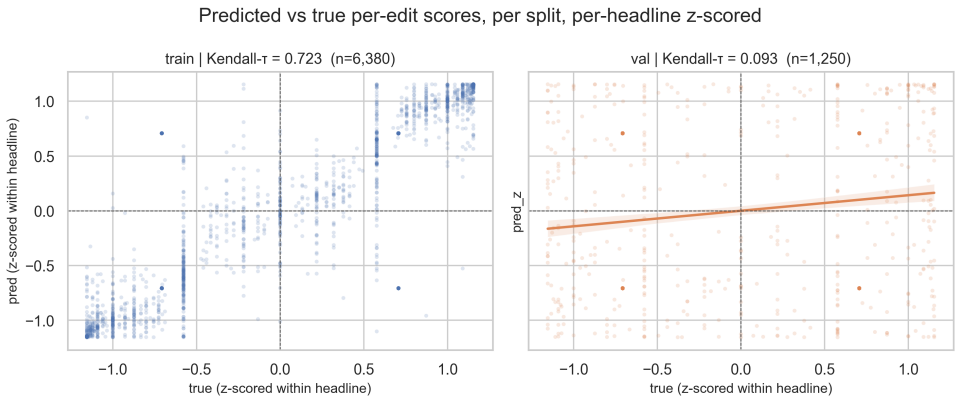


Figure 10: Scatter plot of predicted vs. true funniness scores, z-scored per headline.

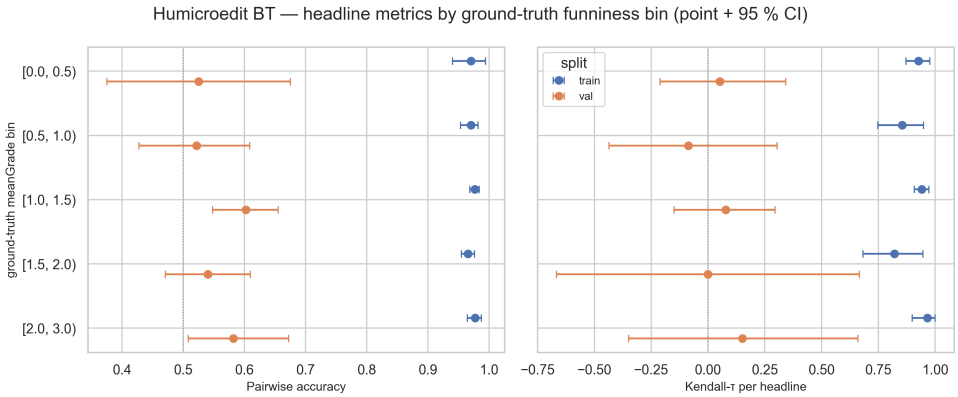


Figure 11: Forest plot of pairwise accuracy and Kendall's τ, for per ground-truth funniness bin.

backbone layers are one way of mitigating overfitting. Although we did not run a clean sweep of frozen backbone layers, we found that freezing the first six of the twelve backbone layers was the only setting that improved validation τ slightly ($0.025\,\tau$ reduction when freezing the backbone entirely, $0.02\,\tau$ reduction when freezing no layers).

Handcrafted features As mentioned in the Theory section, prior work has shown that handcrafted humor-related features can improve humor detection performance. We designed a 34-dimensional feature vector that is concatenated with the headline embeddings before being fed into the MLP head of the model. These features are

- a 10-dimensional emotion feature vector for the edit word and the replacement word each. The emotion feature vector is described in the methods.
- The delta of the emotion feature vectors, $\Delta = e_{\text{edit}} - e_{\text{replacement}}$
- The sum of the Δ vector
- A binary variable that indicates whether the replacement word was present in the emotion lexicon
- A variable measuring the relative length difference between the edit and the replacement word
- The character Jaccard, a numerical measure for the shared vocabulary of characters in strings independent of their order and frequency

This feature vector aims to capture the emotional change induced by the replacement word linearly visible to the MLP head; the out-of-vocabulary, length, and Jaccard capture lexical surprise and flag near-synonym edits. Experimentally, we found that adding these features to the headline embeddings improved validation τ by 0.01 - a small yet useful improvement. To test whether the features captured complementary or overlapping information from the embeddings, we trained a judge with solely the emotion feature vector and an MLP head. This judge scored just 0.01 τ below the baseline. This indicates that the embeddings and the feature capture mostly overlapping information.

4.4.1.2 Qualitative analysis of $\mathcal{J}$ failures

We analyze the 25 validation headlines with the largest absolute error in predicted funniness score to find common failure modes. This is not a representative sample of the distribution; it specifically analyzes failures in the extreme tails.

The main finding is that the model overpredicts funniness when the edit is incongruous but semantically meaningless. In every false-funny (model predicted high, humans rated low) case, the edit is a random concrete noun dropped into a slot where it fits grammatically but produces no hidden meaning, no pun, no mockery, or coherent image. Humans rate these as unfunny while the model scores them as very funny. Consider the headline

```
The Puerto Rican migration could shape Florida politics for
years to come
```

where “Florida” is replaced by “tree”:

```
The Puerto Rican migration could shape tree politics for
years to come
```

Humans rated this with a score of 0.4, the model predicted 2.94. The word replacement word “tree” fits grammatically and creates an incongruency for the reader - the term “tree politics”, even if interpreted as politics regarding trees or forestry, has nothing to do with Puerto Rican migrations. Fittingly, the model underpredicts when the edit is incongruous and meaningful. In these cases, the edit word was usually a pun on another word in the headline, served as a specific cultural reference, or created a more abstract meaning for the headline. For example, the original headline

```
Donald Trump presses a red button on his desk and a butler
brings him a Coke
```

for which “Coke” was changed to “comb” (model predicted 0.23, humans rated 2.6). The edit word fits grammatically, and although it arguably creates less incongruence than the previous example, because a comb is something that could be delivered by a butler, it is funnier because it relies on the common Trump-hair meme, a cultural reference to Donald Trump being obsessed with his unique hairstyle.

To summarize, the humor judge tracks incongruence but not its resolution. Human funniness in this set depends on the edit doing additional work beyond being unexpected, whether a pun uses surrounding words, a coherent absurd scenario, or a cultural reference. The model rewards the surprise of the substitution itself and does not appear to register whether it adds a new layer to the headline.

4.4.2 Generator $\mathcal{G}$

We now analyze how the generator $\mathcal{G}$, trained with the humor reward, behaves. We ask whether the same plateau observed on the NYCCC track recurs here, and whether the limitations of $\mathcal{J}$ surface differently when the output space is restricted to a single replacement word.

4.4.2.1 Quantitative analysis of headline edits

Next, we analyze the edit-words that the system generates after being trained with GRPO. Table 4.17 summarizes the quantitative results of GRPO-generated headline edits on the validation set.

All reward/scores that are reported below are absolute values. Since the humor and the CoVO rewards have different magnitudes, we normalized both terms during training, as described in Section 3.1.2.2.

Temp.	Model	Humor ↑	CoVO Combined ↑	CoVO Value ↑	CoVO Originality ↓	Mixed ↑
0.8	GRPO-HUMOR	**1.226** ± 0.859	3.257 ± 4.549	−6.312 ± 3.205	**-9.569** ± 3.277	4.483 ± 4.521
	SFT	1.202 ± 0.857	3.273 ± 4.527	−6.293 ± 3.202	−9.566 ± 3.262	4.475 ± 4.504
	GRPO-COVO	1.187 ± 0.843	**3.971** ± 4.744	**-6.244** ± 3.225	−10.215 ± 3.442	**5.158** ± 4.722
	GRPO-MIXED	1.186 ± 0.849	3.910 ± 4.713	−6.251 ± 3.222	−10.161 ± 3.423	5.096 ± 4.692
1.2	GRPO-HUMOR	**1.209** ± 0.856	3.180 ± 4.561	−6.357 ± 3.216	**-9.538** ± 3.281	4.390 ± 4.529
	SFT	1.201 ± 0.855	3.159 ± 4.540	−6.329 ± 3.191	−9.487 ± 3.262	4.360 ± 4.511
	GRPO-COVO	1.184 ± 0.844	**3.835** ± 4.729	**-6.298** ± 3.229	−10.133 ± 3.420	**5.019** ± 4.706
	GRPO-MIXED	1.186 ± 0.844	3.797 ± 4.710	−6.289 ± 3.227	−10.086 ± 3.400	4.983 ± 4.684

Table 4.17: Model evaluation results (mean ± std) across reward metrics at temperatures 0.8 and 1.2.

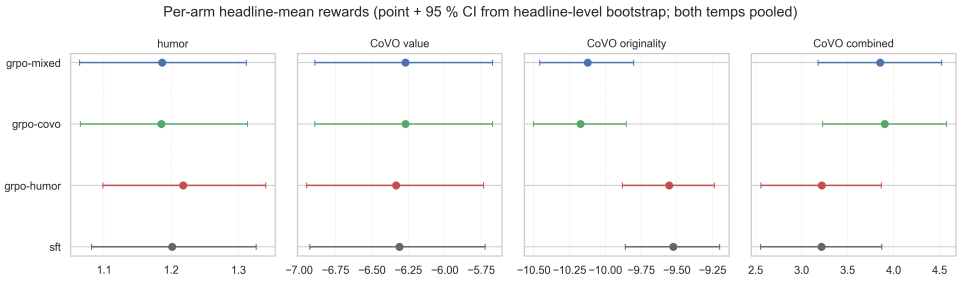


Figure 12: Forest plot of different reward formulations. Mean and 95% CI are shown for different evaluation metrics.

At both temperatures, the $\mathcal{G}_{\text{humor}}$ model wins when evaluated by $\mathcal{J}$ and CoVO originality score. The CoVO-only generator $\mathcal{G}_{\text{CoVO}}$ is the best scorer for the CoVO, CoVO value, and mixed rewards. This is a weak validation of the method: when we use a reward component as a scoring metric, the policy that was used to optimize that reward performs well. Interestingly, $\mathcal{G}_{\text{mixed}}$ wins none of the categories. This could indicate that CoVO and $\mathcal{J}$ rewards are orthogonal, and a policy struggles to optimize both simultaneously.

Figure 12 shows the mean and 95% confidence interval of the different reward arms. The plots reveal that all per-arm differences, except for CoVO originality, are negligible, as their confidence intervals show substantial overlap, with the CoVO and mixed reward arms showing better performance than the humor and SFT arms. We furthermore see in Figure 13 that all reward arms exhibit the same distribution of scores.

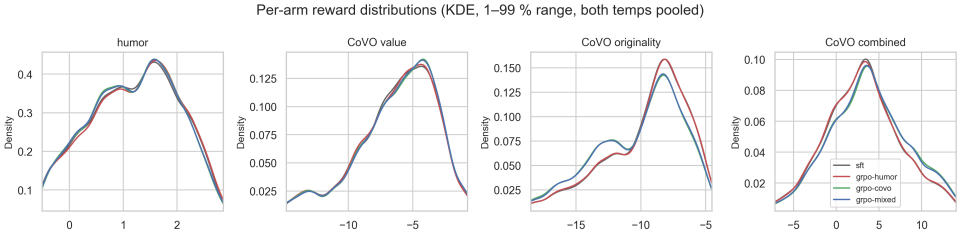


Figure 13: Score distributions of reward arms for different scoring metrics

	SFT	GRPO-Humor	GRPO-CoVO	GRPO-Mixed
SFT	1.00	0.78	0.66	0.68
GRPO-Humor	0.78	1.00	0.69	0.73
GRPO-CoVO	0.66	0.69	1.00	0.94
GRPO-Mixed	0.68	0.73	0.94	1.00

Table 4.18: Fraction of headlines where a pair of reward arms generated the same edit word.

4.4.2.2 Qualitative analysis of headline edits

We analyzed whether the different policies converge on generating the same edit words for a headline. The results (Table 4.18) show that $\mathcal{G}_{\text{CoVO}}$ and $\mathcal{G}_{\text{mixed}}$ produce the same edit on 94% of headlines, while $\mathcal{G}_{\text{humor}}$ and $\mathcal{G}_{\text{SFT}}$ edits overlap on 78% headlines. The reward arms show two behavioral clusters: SFT and $\mathcal{J}$ vs. CoVO and mixed. For example

```
Stephen Hawking warns: Humanity may have less than 600 years
to leave <Earth/>

Earth →
sft: planetarium
grpo-humor: planetarium
grpo-covo: school
grpo-mixed: school
```

The mixed arm is indistinguishable from the CoVO arm, and group-humor barely deviated from its SFT initialization.

We define the type-token ratio as the fraction $\frac{\text{n\_unique}}{\text{n\_total}}$, the proportion of unique edit words. The last row of Table 4.19 shows that SFT has the highest type-token ratio. Furthermore, the arms that scored highest in originality (CoVO, mixed) have a lower ratio. So edit words that were more surprising under the frozen reference coincided with less diverse output, a symptom of reward hacking. The policies could maximize surprise reward by building a small set of words in contexts where the reference model assigns them a low probability. For this task, the CoVO originality reward specifically is a proxy for how improbable a single word is under a frozen policy, and maximizing it led the policy to collapse onto a small subset of words. It seems that the reference model's low probability tokens were not very dependent on the input, i.e. the original headline. Otherwise, we would expect less overlap between headline edits. This is further emphasized by all arms sharing a nearly identical set of top edit words. An overview is given in Table 4.19. A pattern that is apparent is that words related to the Trump-hair joke ('hair', 'trump', 'toupee') are common among all arms. This joke pattern existed in the training set; qualitative inspection showed that the generator arms were able to learn this joke, but frequently applied the corresponding edit words to headlines that are unsuitable for the joke. For example in the headline

Rank	SFT	GRPO-HUMOR	GRPO-COVO	GRPO-MIXED
1	dog (101)	dog (104)	hair (126)	dog (125)
2	hair (82)	hair (89)	dog (123)	hair (124)
3	bathroom (80)	eat (82)	eat (96)	dance (97)
4	eat (80)	dance (81)	party (94)	eat (97)
5	dance (65)	bathroom (80)	trump (94)	party (92)
6	party (65)	puppy (65)	dance (93)	trump (92)
7	trump (65)	cats (61)	bathroom (73)	cats (85)
8	puppy (57)	party (58)	cats (71)	bathroom (69)
9	toupee (56)	trump (57)	car (54)	circus (53)
10	burn (54)	burn (56)	puppy (53)	car (52)
TTR	0.226	0.219	0.208	0.205

Table 4.19: Top ten edit words produced by each policy on the validation generation split, with raw frequencies in parentheses. The lexicon is almost entirely shared across arms; the GRPO arms reuse the same SFT vocabulary at a higher concentration, reflected in the falling type–token ratio (TTR).

```
`German police arrest Syrian suspect, avert major
terrorist attack'
```

all arms replaced ‘attack’ with ‘toupee’ and earned a high humor score from $\mathcal{J}$. In other words, the policies learned to apply this joke blindly.

Summary Work on the Humicroedit data set revealed several findings. The CoVO originality reward was the only metric under which reward arms produced numerically-distinguishable results. A qualitative look at the results showed that arms have a great overlap in edit words. Whenever the originality term was part of the reward, the set of unique edit words decreased; the policy converged on a small set of edit-words that were surprising (but not diverse) under the reference model. The humor judge model showed fatal overfitting symptoms, rendering it essentially unreliable to provide reward gradients. Qualitative analyses revealed here that it rewards headline edits that are incongruous, independent of whether the incongruence was resolved or added a layer of interpretation, a criteria that proved essential for high human ratings.

5

Discussion

In the following, we aggregate all findings from the two dataset tracks and discuss results in the broader context of the existing literature.

Recall the two research questions

RQ1 Does aligning the generator with a composite reward $\mathcal{J}$ + CoVO improve estimated funniness over a fine-tuned baseline?

RQ2 Is the learned humor judge $\mathcal{J}$ a reliable reward signal, and can it capture incongruity resolution?

Based on all results, aligning the generator $\mathcal{G}$ with our composite reward did not improve estimated funniness over the $\mathcal{G}_{\text{SFT}}$ baseline. No tested arm was statistically distinguishable from $\mathcal{G}_{\text{SFT}}$ on automated metrics. The human-preference study echoed this, with participants slightly favoring $\mathcal{G}_{\text{SFT}}$-generated captions over the aligned model.

The ablation experiments locate the failure: the optimizer is healthy, but the reward signal does not move the policy in a useful direction. Across KL, learning rate, group size, alternative algorithms, and a vision-language extension, the plateau persists. The reward, not the optimization step, is the ceiling.

Within the reward, the humor judge $\mathcal{J}$ rewards incongruity but not its resolution; it overfits to tone and statistical patterns rather than to humor mechanisms. The CoVO term, intended to compensate, turns out to be too coarse a proxy for incongruity resolution and, by design, tends to push the policy away from the SFT region. We therefore conclude that the learned humor judge is not a reliable reward signal for alignment, and that information-theoretic surprise without semantic resolution is insufficient for rewarding incongruity resolution.

The remainder of this section explains how these two outcomes arise, working from the proximate cause, the reward rather than the optimizer, to the underlying one, the absence of any signal for why something is funny.

The reward bottleneck. Before attributing this lack of improvement to any single component, the ablations establish where the bottleneck lies: the optimizer itself is healthy, and the reward fails to supply a usable gradient. The ablations identify the

failure in the reward: GRPO optimized a dummy reward, while other optimization algorithms failed in the same way on our composite reward. KL divergence, learning rate, and group size were each ruled out: removing KL did not free the policy, higher LR only produced reward-hacked mode collapse, and larger groups changed nothing. PPO and DPO reproduce the same plateau, confirming that the optimizer is not the limiting factor. The remaining two subsections, therefore, examine the reward's two components in turn.

Incongruity resolution. The judge can detect incongruity, but does not value whether it is resolved. It learned a proxy: surface-level conformity and statistical patterns rather than humor, which arises only from the resolution of incongruity. Humicroedit makes this clearest: edits without resolution scored high, while incongruity with resolution (the Trump-hair/comb jokes in appropriate contexts) scored low; the backbone lacked the world knowledge to connect 'hair' to the cultural Trump-hair meme and only observed that such edits tend to be rated highly. NYCCC shows the same pattern: the judge rewarded style-conforming captions tied to an object in the scene but could not separate generic, weakly grounded captions from those that use the cartoon's central tension to resolve the incongruity. The `@` penalty is a concrete illustration of it latching onto surface features. The reason it learned these proxies is that the training data provides caption-level labels without reasoning, so the model has no signal as to why something is funny.

CoVO granularity. CoVO was intended to compensate by rewarding surprising-but-relevant generations; our results show it was structurally working against this objective. The originality term punishes the high-quality SFT region by design: a caption that is surprising (low probability) under the SFT reference may simply be a poor caption, unlike the curated training captions, so the term pushes GRPO away from SFT-style output. The value term cannot prevent this; it only punishes captions unrelated to the cartoon description. The base-reference ablation hits the same issues, which rules out the "merely displaced from SFT" failure mode and strengthens the point: CoVO is too coarse a representation of incongruity. The Humicroedit replacement words collapse onto a small set of improbable but repetitive words, a clean demonstration of what CoVO actually optimizes.

Policy saturation. These weaknesses would be tolerable if the generator had room to improve, but it starts from a near-saturated point. The SFT checkpoint was trained on the top 10% of human NYCCC captions, so it already produces plausible, format-correct captions: a strength as a warm start, a weakness because it leaves GRPO little room to move. The judge is least reliable in exactly this high-quality region. Our per-bin accuracy analysis highlighted that ranking the actually-funny captions is hardest, so the reward provides no usable gradient where the policy already lives. The generator's ceiling is therefore the interaction between a near-saturated policy and a reward that is weakest precisely where the policy starts.

The judge's blindness to resolution, CoVO's coarseness, and the policy's saturation share one underlying cause: none of these components has access to the world

knowledge, cultural references, and contextual grounding that determine whether something is funny. We implicitly assumed Qwen3-4B had enough world knowledge baked in through its pretraining to draw on during generation. This assumption was at least partially wrong, and neither dataset supplied the missing context. Neither includes reasoning steps or explanations of why an output is funny. Visual grounding is part of this gap: many cartoons carry intricate details absent from the text description, which a text-only decoder cannot see. However, our multimodal humor judge was unable to score captions accurately despite having access to the images. The user study lends further support: participants asked for more cultural references and stronger image grounding, and inter-rater agreement was lowest for captions that required relevant background knowledge to resolve the incongruity humorously (raters who got the reference agreed on the human caption; others did not). What this implies is that a useful reward must articulate why an output is funny rather than merely rank it; we continue this in the future directions below. Both components of our system, the humor judge and the humor generator, fail because they lack the context of why something is funny or not to a human. What is needed is a reward that can articulate the score, not just rank outcomes. This points towards reasoning-capable judges and generators. Wang et al. [2024] and Zhong et al. [2024] supersede a chain-of-thought approach with their "Creative Leap-of-Thought" framework in which a model is trained to first explore semantically distant concepts and then refine and filter these connections to encourage creative thinking rather than a linear CoT. However, this framework is only suitable for generating creative joke ideas, not reasoning about the quality of a joke if all contextual information is available. The problem can be composed into two steps. First, both the judge and the generator require context about the joke's setup. In the case of NYCCC, this includes correctly interpreting all entities in the cartoon, reasoning about their relationships, noticing visual subtleties in the image, and drawing on background information. This poses the fundamental problem of building language models with world knowledge equivalent to that of humans. Another solution is to equip the generator and judge with the tools to gather contextual knowledge on the fly. This could be done by using curated databases or by giving the models a search engine tool to draw information into context that they believe is needed to judge or generate a joke in that context.

5.1 Comparison to previous work

We have several results from previous work that are directly comparable to ours. The following compares our findings to three reference points: the NYCCC contest benchmark and its associated cartoon-captioning systems, the Humicroedit benchmark and its competition submissions, and the original CoVO study from which we adopted the auxiliary reward.

5.1.1 The NYCCC contest

In their paper introducing the NYCCC dataset and cartoon captioning task, Zhang et al. [2024] report a pairwise accuracy of 67% for their best-performing evaluation

model. This is just slightly above the 95% confidence interval for our humor judge. Zhang et al. utilized GPT4-Turbo with 5-shot in-context learning, while our method achieves comparable performance using a lightweight open-source model. Zhang et al. report win rates of model-generated captions vs human-written captions, evaluated by their best evaluation model. Their RL- or preference-fine-tuned models beat human captions between 9 and 35% of the time, depending on which relative rank (e.g., top 10) the human captions were chosen from. Our experimental setup does not allow for a direct comparison between our win rate and theirs. Even though our judge models report very similar pairwise accuracies, we do not know whether they suffered from similar collapses to ours, which could be masked by the metric. Furthermore, Zhang et al. compute win-rates on their group comparison metric (deciding whether a group of model-generated captions is funnier than a group of human captions), whereas we use pairwise comparisons.

As we outlined in the Ablations section, we found that captions generated by the VLM were inferior (numerically and qualitatively) to captions generated based on textual descriptions. Zhang et al. report the same finding.

Zhang et al. also used RL and DPO in an attempt to fine-tune their generation models. Their findings align with ours - even if the optimization algorithm can optimize the generator to produce captions that *score* higher, the outputs do not necessarily improve for humans.

In their qualitative analysis of generated captions, Zhang et al. name very long captions, missing image-groundedness, and missing levels of possible interpretations as the main failure modes of the models. This aligns well with our qualitative findings.

5.1.2 The Humicroedit dataset

Moving from cartoon captioning to headline editing, our results on Humicroedit can be compared to the original benchmark associated with the dataset.

When Hossain et al. introduced the Humicroedit dataset as a benchmark, they collected submissions from different teams for the task of ranking pairs of headline edits. According to their results, the best team achieved an accuracy of 67.43%. This is significantly better than our judge's pairwise accuracy of 56.8%. It should be noted that submissions to the Hossain et al. [2020a] competition were highly engineered systems. The winner built an ensemble of 700 unique pretrained language models. An approach of this complexity was out of scope for this thesis, especially since the judge $\mathcal{J}$ model only served as a reward model in our system. Moreover, Hossain et al. did not report how reliably their models could be used to rank groups of headline edits.

In summary, our method achieves performance comparable to that of Zhang et al. [2024] while exhibiting the same collapses and caption-quality issues.

5.1.3 The CoVO study

Beyond the dataset-specific benchmarks, we compare our results with the CoVO reward to the study that introduced it, since our results differ significantly.

When Franceschelli and Musolesi derived the CoVO score as a numeric reward that measures value and originality, it was used to fine-tune LLMs on three different tasks (see Section 2.6). On all three tasks (poem writing, math problem solving, and `NoveltyBench`), the non-finetuned models performed relatively well, i.e., they were already able to perform the task. CoVO-training improved the performance by increasing the model's output diversity without sacrificing quality. This stands in sharp contrast to our experimental setup. Both, Qwen3-4B and $\mathcal{G}_{\text{SFT}}$, were not able to perform either of the humor generation tasks at a decent level. Aligning them using GRPO and a humor + CoVO reward did not improve their performance, but this failure cannot be attributed to the CoVO itself.

6

Conclusion

This chapter summarizes our work, presents our contributions and the study's limitations, and provides directions for future research.

We fine-tuned a decoder model for two humor generation tasks using SFT and GRPO. The reward consisted of a human-preference-based reward model and an information-theoretic component that rewarded surprising-but-relevant outputs. The results of the experiments show that our reward does not quantitatively improve the outputs relative to the SFT baseline. Policies were either indistinguishable from SFT or exhibited reward hacking, indicated by degenerate outputs. Qualitatively, the generated outputs failed to engage with the task's context: cartoon captions did not leverage the cartoon's central tension. Edited headlines were surprising, but did not allow for a humorous interpretation. The small-scale human study found that users vastly prefer human-written cartoon captions over generated ones. Our ablations locate this failure in the reward signal rather than the optimization step. The humor judge $\mathcal{J}$ rewards incongruity but not its resolution, overfitting to shallow signals and failing to recognize *why* some jokes are funny only in a certain context. Both components ultimately lack access to the world knowledge, cultural references, and contextual grounding that humans require to determine whether something is funny, a gap that neither dataset fills, and that a scalar reward cannot close.

6.1 Contributions

This work contributes to the existing literature on the four objectives outlined in Chapter 1. We formulated a composite reward to align a language model on two humor-generation tasks. Not only was operationalizing incongruity-resolution in humor a new domain for the CoVO reward term, but we also extended the framework from text tokens to image tokens, showed that the extension is information-theoretically equivalent to the original formulation, and demonstrated its practical use for vision-language humor tasks. Systematic ablations on the generator model isolated the humor judge's inability to detect incongruity resolution as the bottleneck, rather than the optimizer or the CoVO term.

Apart from methodological contributions, our cartoon-captioning results reproduce Zhang et al.'s qualitative and quantitative findings where the approaches are comparable, and extend their RL comparison by testing GRPO and reaching the same

diagnosis. Automated evaluation in our analysis was complemented by a descriptive coding analysis of 700 generated captions and a small-scale human-preference study, both of which converge on image grounding and cultural reference as the missing ingredients.

The experiments confirm that humor tasks, whether generation or evaluation, are inherently difficult and that continuous scalar metrics struggle to capture them reliably. By discussing these findings in their broader context, we provide concrete recommendations for future research in this field.

6.2 Limitations

One bias runs through this whole study. Humor depends heavily on cultural context; what we find funny depends not only on our individual preferences but also on our language, knowledge, and social norms, all of which are embedded in that context. Many aspects of this work have been carried out in the context of Western culture. The datasets for both of our tasks have been collected and published by research teams affiliated with universities in the USA; participants in the user study had primarily Western backgrounds and lived in Western and Northern Europe. This cultural bias impacts our methodological approach, the results (since the models were trained on Western-influenced data), and the interpretation of results.

Our results indicated that metrics can be unreliable for capturing the full depth of humor. Yet during the development of this work, we had to make many decisions about the performance of intermediate models. For this, we relied on a combination of metrics and qualitative evaluation of model output. A structured and rigorous approach to the qualitative evaluation, as we had for our final evaluation, would have been unfeasible given the time constraints. Rather, evaluations always were judgment calls based on the model output. This makes the final configuration of our system biased towards our cultural and personal humor preferences. The same bias appears in the qualitative coding approach that we utilized. The single coder for all responses is one of the authors. His preferences influenced the selection of joke patterns and coding decisions reflected in our results.

Cultural bias in a topic that is as subjective as humor is hard to avoid. However, it is important to interpret this study with the biases in mind, as results may not generalize to other cultures or even other individuals. To assess properties of humors that generalize cross-culturally, future work should attempt to have both, diverse research teams and datasets from different cultural origins.

Next to the external limitations, such as cultural bias, this work has other limitations. The way we operationalize humor and model it through incongruity resolution is narrow and just one of many valid ways. We refer to the limitations section of Franceschelli and Musolesi [2025] for an extensive explanation of CoVO's limitations for modeling surprise and relevance. Furthermore, we cannot rule out the possibility that our negative results are due to poor hyperparameter choices. We ablated and searched many of the knobs of our system, but this was by no means exhaustive.

Furthermore, our results do not support strong claims about the effectiveness of RL for humor generation in general. Only our system, comprised of a generator model trained with a specific reward model, can be said to perform poorly. It is possible that other reward functions exist that better support RL for humor generation. Connected to this is the fact that we tested only one model family for generation: Qwen3. Previous work has successfully used these models for similar tasks, and we have no concrete reason to believe that Qwen3 models are structurally worse at humor tasks than comparable models. Yet, this limits the generalizability of our results to other model families. Lastly, both datasets used were public. Thus, we cannot rule out the possibility that they contaminated the models' pretraining data, thereby affecting performance.

6.3 Future directions

Although our composite reward did not improve generation quality, the negative results clarify where the bottleneck lies. Our work strengthens the field's emerging view that humor generation and evaluation are challenging tasks, even with state-of-the-art language models. Based on our results, we believe that there are two promising approaches to improve humor generation tasks. As stated earlier, most humor datasets lack the contextual information or reasoning trace that explains why some output is funny. Without it, evaluation models have to rely on superficial cues. Future work should investigate how this information can be supplied to the models, for example, in the form of recorded reasoning traces or search tool use. Secondly, we believe that vision-language integration is essential for the cartoon captioning task: images often contain details essential to understanding the cartoon's tension that are not mentioned in the text description. Our experiment using Qwen3-VL for generation was unsuccessful. However, a contrastive model (BLIP-2) as a humor judge showed better performance than a text-only backbone. Future work should focus on improving VL models for humor generation.

Our results provide further support for the field's consensus that rating humor using continuous scores is not feasible with current methods. As others have before, we found that replacing direct regression approaches with pairwise comparisons simplifies the task for the model.

6.4 Concluding remarks

This study showed that training a model for humor generation with GRPO is difficult, as our reward was susceptible to reward hacking and unable to capture the depth of humor. We believe the biggest challenge for this task is providing the generator and reward models with all the context humans use to come up with and judge humor. This can be approached as both an engineering problem and a theoretical gap between psycho-linguistic humor theories and their operationalization in AI systems. Yet, humor remains an interesting lens through which we can study the current ceilings and limitations of large language models. If asked for a recommendation, we

would advise aspiring comedians to pursue this career — at least for now, it remains one of the few AI-safe careers.

Bibliography

Khalid Alnajjar and Mika Hämäläinen. When a computer cracks a joke: Automated generation of humorous headlines. *arXiv preprint arXiv:2109.08702*, 2021.

Miriam Amin and Manuel Burghardt. A survey on approaches to computational humor generation. In Stefania DeGaetano, Anna Kazantseva, Nils Reiter, and Stan Szpakowicz, editors, *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 29–41, Online, December 2020a. International Committee on Computational Linguistics. URL `https://aclanthology.org/2020.latechclfl-1.4/`.

Miriam Amin and Manuel Burghardt. A survey on approaches to computational humor generation. In *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 29–41, 2020b.

Issa Annamoradnejad and Gohar Zoghi. Colbert: Using bert sentence embedding in parallel neural networks for computational humor. *Expert Systems with Applications*, 249:123685, 2024.

Margaret A Boden. *The creative mind: Myths and mechanisms*. Routledge, 2004.

G. Bouma. Normalized (pointwise) mutual information in collocation extraction. 2009.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Yang Chen, Chong Yang, Tu Hu, Xinhao Chen, Man Lan, Li Cai, Xinlin Zhuang, Xuan Lin, Xin Lu, and Aimin Zhou. Are u a joke master? pun generation via multi-stage curriculum learning towards a humor llm. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 878–890, 2024.

Hugging Face. Dataset formats and types. `https://huggingface.co/docs/trl/en/dataset_formats`, 2024. TRL (Transformer Reinforcement Learning) Documentation.

Mark A Ferguson and Thomas E. Ford. Disparagement humor: A theoretical and empirical review of psychoanalytic, superiority, and social identity theories. 21:283 – 312, 2008. doi: 10.1515/humor.2008.014.

Giorgio Franceschelli and Mirco Musolesi. Thinking outside the (gray) box: A context-based score for assessing value and originality in neural text generation. *arXiv preprint arXiv:2502.13207*, 2025.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Christian F Hempelmann, Julia Rayz, Tiansi Dong, and Tristan Miller. Proceedings of the 1st workshop on computational humor (chum). In *Proceedings of the 1st Workshop on Computational Humor (CHum)*, 2025.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL `http://arxiv.org/abs/1606.08415`.

Jack Hessel, Ana Marasović, Jena D Hwang, Lillian Lee, Jeff Da, Rowan Zellers, Robert Mankoff, and Yejin Choi. Do androids laugh at electric sheep? humor "understanding" benchmarks from the new yorker caption contest. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–714, 2023.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. doi: 10.5281/zenodo.1212303.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. Semeval-2020 task 7: Assessing humor in edited news headlines, 2020a. URL `https://arxiv.org/abs/2008.00304`.

Nabil Hossain, John Krumm, Tanvir Sajed, and Henry Kautz. Stimulating creativity with funlines: A case study of humor generation in headlines. In *Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations*, pages 256–262, 2020b.

Zhongzhan Huang, Shanshan Zhong, Pan Zhou, Shanghua Gao, Marinka Zitnik, and Liang Lin. A causality-aware paradigm for evaluating creativity of multimodal large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

EunJeong Hwang, Peter West, and Vered Shwartz. Bottlehumor: Self-informed humor explanation using the information bottleneck principle. *arXiv preprint arXiv:2502.18331*, 2025.

Mete Ismayilzada, Debjit Paul, Antoine Bosselut, and Lonneke van der Plas. Creativity in ai: Progresses and challenges. *arXiv preprint arXiv:2410.17218*, 2024.

Mete Ismayilzada, Antonio Laverghetta, Simone A Luchini, RN Patel, Antoine Bosselut, Lonneke Van Der Plas, and Roger E Beaty. Creative preference optimization. *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9580–9609, 2025.

Antonios Kalloniatis and Panagiotis Adamidis. Computational humor recognition: a systematic literature review. *Artificial Intelligence Review*, 58(2):43, 2024.

Immanuel Kant. *Critique of judgment*. Hackett Publishing, 1987.

Justine T Kao, Roger Levy, and Noah D Goodman. A computational model of linguistic humor in puns. *Cognitive science*, 40(5):1270–1285, 2016.

Hasindu Kariyawasam, Amashi Niwarthana, Alister Palmer, Judy Kay, and Anusha Withana. Appropriate incongruity driven human-ai collaborative tool to assist novices in humorous content generation. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, 2024. doi: 10.1145/3640543.3645161.

Patricia Keith-Spiegel. Early conceptions of humor: Varieties and issues. *The psychology of humor: Theoretical perspectives and empirical issues*, 2:4–39, 1972.

Sean Kim and Lydia B Chilton. Ai humor generation: Cognitive, social and creative skills for effective humor. *arXiv preprint arXiv:2502.07981*, 2025.

Michal Kolomaznik, Vladimír Petrík, M. Sláma, and V. Juřík. The role of socio-emotional attributes in enhancing human-ai collaboration. *Frontiers in Psychology*, 15, 2024. doi: 10.3389/fpsyg.2024.1369957.

Jens Lemmens and Victor De Marez. Computational humor modeling: A survey on the state of the art. *ACM Computing Surveys*, 58(7):1–37, 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL `https://arxiv.org/abs/2301.12597`.

A Peter McGraw and Caleb Warren. Benign violations: Making immoral behavior funny. *Psychological science*, 21(8):1141–1149, 2010.

Saif M Mohammad and Peter D Turney. Nrc emotion lexicon. *National Research Council, Canada*, 2:234, 2013.

Andreea Niculescu, Betsy Van Dijk, Anton Nijholt, Haizhou Li, and Swee Lan See. Making social robots more attractive: the effects of voice pitch, humor and empathy. *International journal of social robotics*, 5(2):171–191, 2013.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, S. Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, M. Simens, Amanda Askell, Peter Welinder, P. Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022. doi: 10.52202/068431-2011.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL `http://arxiv.org/abs/1912.01703`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL `https://arxiv.org/abs/2103.00020`.

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.

Graeme Ritchie. Developing the incongruity-resolution theory. Technical report, 1999.

Ritsu Sakabe, Hwichan Kim, Tosho Hirasawa, and Mamoru Komachi. Assessing the capabilities of llms in humor: A multi-dimensional analysis of oogiri generation and evaluation. *arXiv preprint arXiv:2511.09133*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL `http://arxiv.org/abs/1707.06347`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL `https://arxiv.org/abs/2402.03300`.

Changhao Song, Yazhou Zhang, Hui Gao, Ben Yao, and Peng Zhang. Large language models for subjective language understanding: A survey, 2025. URL `https://arxiv.org/abs/2508.07959`.

Jerry M Suls. A two-stage model for the appreciation of jokes and cartoons: An information-processing analysis. *The psychology of humor: Theoretical perspectives and empirical issues*, 1:81–100, 1972.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Mor Turgeman, Chen Shani, and Dafna Shahaf. One joke to rule them all? on the (im) possibility of generalizing humor. *arXiv preprint arXiv:2508.19402*, 2025.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec.

TRL: Transformers Reinforcement Learning, March 2020. URL https://github.com/huggingface/trl.

Han Wang, Yilin Zhao, Dian Li, Xiaohan Wang, Gang Liu, Xuguang Lan, and Hui Wang. Innovative thinking, infinite humor: Humor research of large language models through structured thought leaps. *arXiv preprint arXiv:2410.10370*, 2024.

Yan Wang, Wei Song, Wei Tao, Antonio Liotta, Dawei Yang, Xinlei Li, Shuyong Gao, Yixuan Sun, Weifeng Ge, Wei Zhang, et al. A systematic review on affective computing: Emotion models, databases, and recent advances. *Information Fusion*, 83:19–52, 2022.

Orion Weller, Nancy Fulda, and Kevin Seppi. Can humor prediction datasets be used for humor generation? humorous headline generation via style transfer. In Beata Beigman Klebanov, Ekaterina Shutova, Patricia Lichtenstein, Smaranda Muresan, Chee Wee, Anna Feldman, and Debanjan Ghosh, editors, *Proceedings of the Second Workshop on Figurative Language Processing*, pages 186–191, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.figlang-1.25. URL https://aclanthology.org/2020.figlang-1.25/.

E. Wilcox, Tiago Pimentel, Clara Meister, and Ryan Cotterell. An information-theoretic analysis of targeted regressions during reading. *Cognition*, 249:105765, 2024. doi: 10.1016/j.cognition.2024.105765.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics, October 2020. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weiling Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. pages 54983–54998, 2024. doi: 10.48550/arxiv.2404.10719.

Xing'an Xu and Juan Liu. Artificial intelligence humor in service recovery. *Annals of Tourism Research*, 95:103439, 2022.

Jiajun Zhang, Shijia Luo, Ruikang Zhang, and Qi Su. Humorchain: Theory-guided multi-stage reasoning for interpretable multimodal humor generation. *arXiv preprint arXiv:2511.21732*, 2025.

Jifan Zhang, Lalit Jain, Yang Guo, Jiayi Chen, Kuan Zhou, Siddharth Suresh, Andrew Wagenmaker, Scott Sievert, Timothy T Rogers, Kevin G Jamieson, et al. Humor in ai: Massive scale crowd-sourced preferences and benchmarks for cartoon captioning. *Advances in Neural Information Processing Systems*, 37:125264–125286, 2024.

Mingming Zheng, Kaizhong Jiang, Ranhui Xu, and Lulu Qi. An adaptive lda optimal topic number selection method in news topic identification. *IEEE Access*, 11: 92273–92284, 2023. doi: 10.1109/access.2023.3308520.

Shanshan Zhong, Zhongzhan Huang, Shanghua Gao, Wushao Wen, Liang Lin, Marinka Zitnik, and Pan Zhou. Let's think outside the box: Exploring leap-of-thought in large language models with creative humor generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13246–13257, 2024.

Kuan Lok Zhou, Jiayi Chen, Siddharth Suresh, Reuben Narad, Timothy T Rogers, Lalit K Jain, Robert D Nowak, Bob Mankoff, and Jifan Zhang. Bridging the creativity understanding gap: Small-scale human alignment enables expert-level humor ranking in llms. *arXiv preprint arXiv:2502.20356*, 2025.

A

Hyperparameters

Humicroedit humor judge

- **Model**: microsoft/deberta-v2-base; problem_type=regression; freeze_n_backbone_layers=6; hidden_dropout_prob=0.1
- **Features**: use_sentiments=true, use_incongruity_features=true (34-d = 20-d NRC EmoLex + 14-d incongruity: emotion delta, OOV, length/char features)
- **Head**: hidden_dims=[], dropout=0.2, activation=gelu
- **Data**: dataset=humicroedit, intra-headline pairs, min_grade_margin=0.0 (all 4018 humicroedit+funlines pairs), include_external_bt=false
- **Training**: loss_type=bradley_terry, epochs=20, batch_size=16, grad_accum=2 (eff. bs=32), eval_batch_size=128 (cluster), pairwise=true
- **BT loss**: margin_scaling=0.4, $\lambda_{\text{MSE}} = 0.5$
- **Optimizer**: lr_backbone=2e-5, lr_head=1e-4, warmup_ratio=0.06, weight_decay=0.01
- **Best-model metric**: pairwise_accuracy (greater_is_better)
- **Early stopping**: disabled
- **Precision**: bf16=true; **Seed**: 42

Humicroedit Supervised Fine-Tuning

- **Model**: Qwen/Qwen3-4B-Instruct-2507
- **LoRA**: r=32, $\alpha = 64$, dropout=0.05, targets = q,k,v,o,gate,up,down_proj
- **Data**: dataset=humicroedit, max_length=78, include_unfun_sft=false (pure humicroedit+funlines)
- **Training**: epochs=4, batch_size=8, per_device_batch_size=1, grad_accum=8 (eff. bs=8), lr=2e-5, weight_decay=0.01, bf16=true, fp16=false, max_seq_length=256, eval_steps=200, save_steps=500, early_stopping(patience=3,

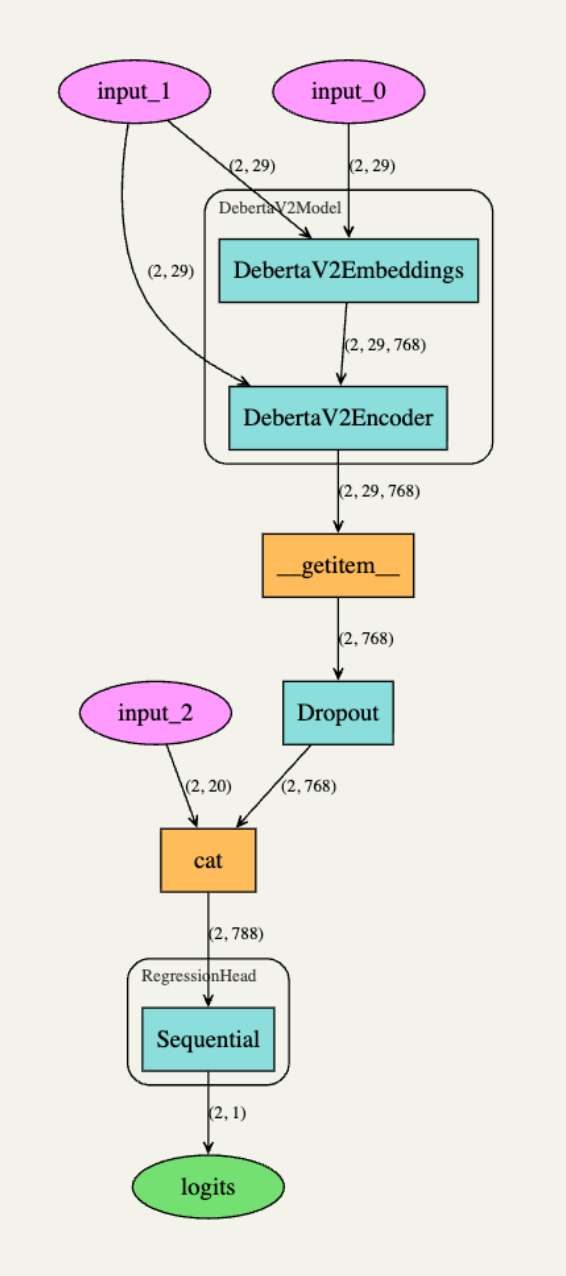


Figure 1: Architecture of the humor regressor. Input_1 is a batch of tokenized headlines, input_0 is a batch of corresponding attention masks, and input_2 is a batch of computed emotionality features.

threshold=0.01)

- **Generation**: temperature=0.7, top_p=0.9, num_return_sequences=1, max_new_tokens=10 (single-word edit)
- **Seed**: 42

Humicroedit GRPO (shared base for all three arms)

- **Generator (policy)**: Qwen/Qwen3-4B-Instruct-2507, 4-bit (bitsandbytes), LoRA r=32, $\alpha = 64$, dropout=0.05, targets = q,k,v,o_proj
- **Evaluator (frozen)**: humicroedit DeBERTa scorer (§A); `regressor_checkpoint=best.pt`
- **Generator initialization**: humicroedit SFT base (§A)
- **Data**: dataset=humicroedit, use_all_content_words=false, use_images=false, evaluator_uses_images=false
- **Training**: group_size=8, per_device_batch_size=2, grad_accum=4 (eff. bs=8), lr=5e-7, kl_weight=0.04, max_completion_length=10 (single-word edit), temperature=0.8, eval_steps=25, eval_n_random=10, eval_n_trajectory=10, save_steps=500, save_total_limit=2, regressor_dtype=float32, regressor_compile=false, regressor_batch_size=16
- **GRPO clipping**: $\varepsilon_{\text{low}} = 0.2$, $\varepsilon_{\text{high}} = 0.28$ (clip-higher); reward_normalization.enabled=true, clip=5.0
- **Reward (shared)**: mode=covo, pos_penalty_weight=0.0 (zeroed for cross-track parity with NYCCC), momentum=0.05, normalize=false, humor_normalize=false, target_mean=0.5, target_std=0.25, covo_ref_mode=sft
- **Precision**: bf16=true; **Seed**: 42; **auto_cleanup**: false

Humicroedit GRPO (humor-only)

- **Inherits**: shared humicroedit GRPO base (§A)
- **Reward weights**: humor_weight=**1.0**, covo_weight=0.0, $\lambda_{\text{value}} = 1.0$, $\lambda_{\text{orig}} = 1.0$ (CoVO λ's inert at covo_weight=0)

Humicroedit GRPO (CoVO-only)

- **Inherits**: shared humicroedit GRPO base (§A)
- **Reward weights**: humor_weight=0.0 (no humor signal in gradient), covo_weight=**1.0**, $\lambda_{\text{value}} = \mathbf{1.0}$, $\lambda_{\text{orig}} = \mathbf{1.0}$

Humicroedit GRPO (mixed humor+CoVO)

- **Inherits**: shared humicroedit GRPO base (§A)
- **Reward weights**: humor_weight=**0.5**, covo_weight=**0.5**, $\lambda_{\text{value}} = \mathbf{1.0}$, $\lambda_{\text{orig}} = \mathbf{0.5}$

NYCCC Supervised Fine-Tuning

- **Model**: Qwen/Qwen3-4B-Instruct-2507 (text-only; 'use_images=false')
- **LoRA**: r=32, $\alpha = 64$, dropout=0.05, targets = q,k,v,o,gate,up,down_proj
- **Data**: dataset=nyccc, max_length=78, top_percent=0.1
- **Training**: epochs=4, batch_size=8, per_device_batch_size=1, grad_accum=8 (eff. bs=8), lr=2e-5, weight_decay=0.01, bf16=true, fp16=false, max_seq_length=512, eval_steps=200, save_steps=500, early_stopping(patience=3, threshold=0.01)
- **Generation**: temperature=0.7, top_p=0.9, num_return_sequences=1, max_new_tokens=30
- **Seed**: 42

NYCCC Humor judge

- **Multimodal model**: Salesforce/blip2-opt-2.7b (freeze_vision=true, freeze_language_model=true; only Q-Former and regression head train)
- **Head**: hidden_dims=[], dropout=0.1, activation=gelu
- **Problem type**: regression; max_length=128; hidden_dropout_prob=0.1
- **Data**: dataset=nyccc, use_images=true, nyccc_min_stds=1.5, nyccc_pairs_per_contest=100
- **Training**: loss_type=bradley_terry, epochs=10, batch_size=4, grad_accum=4 (eff. bs=16), eval_batch_size=8, pairwise=true
- **BT loss**: margin_scaling=0.4, $\lambda_{\text{MSE}} = 0.5$
- **Optimizer**: lr_backbone=5e-5, lr_head=1e-4, warmup_ratio=0.06, weight_decay=0.01
- **Best-model metric**: kendall_tau_b (greater_is_better)
- **Precision**: bf16=true

NYCCC GRPO (shared base config for all five GRPO runs)

- **Generator (policy)**: Qwen/Qwen3-4B-Instruct-2507, 4-bit (bitsandbytes), LoRA r=32, $\alpha = 64$, dropout=0.05, targets = q,k,v,o_proj

- **Evaluator (frozen)**: BLIP-2 multimodal humor judge
- **Generator initialization**: SFT checkpoint of Qwen3-4B-Instruct (frozen)
- **Data**: dataset=nyccc, top_percent=0.1, use_images=false (text-only generator), evaluator_uses_images=true
- **Training**: group_size=8, per_device_batch_size=2, grad_accum=4 (eff. bs=8), lr=5e-7, max_steps=3000 (epochs=1 ignored), kl_weight=0.04, max_completion_length=40, temperature=0.8, eval_steps=25, eval_n_random=10, eval_n_trajectory=10, save_steps=500, save_total_limit=2, regressor_dtype=float32, regressor_compile=false, regressor_batch_size=16
- **GRPO clipping**: $\varepsilon_{\text{low}} = 0.2$, $\varepsilon_{\text{high}} = 0.28$ (clip-higher)
- **Reward (shared)**: pos_penalty_weight=0.0, momentum=0.05, normalize=false, target_mean=0.5, target_std=0.25, covo_ref_mode=sft
- **Precision**: bf16=true; **Seed**: 42

NYCCC GRPO (humor-only reward)

- **Inherits**: shared GRPO base (§A)
- **Reward weights**: humor_weight=**1.0**, covo_weight=0.0, $\lambda_{\text{value}} = 1.0$, $\lambda_{\text{orig}} = 1.0$ (CoVO λ's inert at covo_weight=0)

NYCCC GRPO (CoVO value-only)

- **Inherits**: shared GRPO base (§A)
- **Reward weights**: humor_weight=0.0 (eval-only), covo_weight=**1.0**, $\lambda_{\text{value}} =$ **1.0**, $\lambda_{\text{orig}} =$ **0.0**

NYCCC GRPO (CoVO originality-only)

- **Inherits**: shared GRPO base (§A)
- **Reward weights**: humor_weight=0.0 (eval-only), covo_weight=**1.0**, $\lambda_{\text{value}} =$ **0.0**, $\lambda_{\text{orig}} =$ **1.0**

NYCCC GRPO (CoVO combined)

- **Inherits**: shared GRPO base (§A)
- **Reward weights**: humor_weight=0.0 (eval-only), covo_weight=**1.0**, $\lambda_{\text{value}} =$ **1.0**, $\lambda_{\text{orig}} =$ **1.0**

NYCCC GRPO (mixed humor+CoVO)

- **Inherits**: shared GRPO base (§A)
- **Reward weights**: humor_weight=**0.5**, covo_weight=**0.5**, $\lambda_{\text{value}} = \mathbf{1.0}$, $\lambda_{\text{orig}} = \mathbf{0.5}$

PPO hyperparameters

- **Policy / generator**: Qwen/Qwen3-4B-Instruct-2507, loaded in 4-bit (bitsandbytes `load_in_4bit=true`), initialized from SFT adapter A.
- **LoRA**: $r = 32$, $\alpha = 64$, dropout = 0.05, target modules $\{q, k, v, o\}_{\text{\_proj}}$.
- **Reward model (frozen)**: Humor judge A; `regressor_dtype=float32`, `regressor_batch_size=16`, `regressor_compile=false`.
- **Data**: NYCCC, `top_percent=0.1`, text-only policy (`use_images=false`), multimodal evaluator (`evaluator_uses_images=true`).
- **Batching**: `per_device_train_batch_size=4`, `gradient_accumulation_steps=16` (effective $4 \times 16 = 64$ prompts / step — sample-count parity with the GRPO baseline $2 \times 4 \times 8$).
- **Optimization**: lr = 5e−7, weight decay 0.01, bf16, epochs = 1, `max_steps`= 5000.
- **Rollout**: `max_completion_length=40`, temperature 0.8.
- **PPO core**: cliprange $\varepsilon = 0.2$, value cliprange 0.2, v_f coefficient 0.1, discount $\gamma = 1.0$, GAE $\lambda = 0.95$, `num_ppo_epochs=1`, `num_mini_batches=1`.
- **KL**: `kl_weight`= 0.04, estimator k_1, `whiten_rewards=false` (reward composer already per-component normalizes).
- **Value head**: random init (TRL default), initializer range 0.2, summary dropout default.
- **Reward composition**: `humor_weight`= 0.5, `covo_weight`= 0.5, $\lambda_{\text{value}} = 1.0$, $\lambda_{\text{orig}} = 1.0$, `pos_penalty_weight`= 0.0, EMA momentum 0.05, per-component normalise (`target_mean`= 0.5, `target_std`= 0.25), `covo_ref_mode=sft`.
- **Eval**: every 50 steps; `eval_n_random=10`, `eval_n_trajectory=10`.
- **Checkpointing**: `save_steps=200`, `save_total_limit=2`.
- **Seed**: 42.

Iterative DPO hyperparameters

- **Policy / generator**: Qwen/Qwen3-4B-Instruct-2507, initialized from the same SFT adapter A.

- **LoRA**: $r = 16$, $\alpha = 32$, dropout $= 0.05$, target modules $\{q, k, v, o\}$_proj (lighter than PPO/GRPO to limit overfitting on small preference sets).
- **Reference policy**: frozen SFT across all iterations (does *not* track the trained policy).
- **Reward model for preference labelling (frozen)**: Humor judge A; `regressor_dtype=float32`, `regressor_batch_size=16`.
- **DPO loss**: $\beta = 0.1$ (winner of the $\{0.01, 0.1, 0.3, 0.5\}$ sweep on iter 1).
- **Iterative meta-loop**: `num_iterations=3`; at each iter k sample N captions/-cartoon from current policy G_{k-1}, rank by humor+CoVO z, build matched-rank preferences, train against frozen SFT ref, merge adapter into base for G_k.
- **Per-iter preference-data build**: `n_candidates_per_cartoon=50`, `top_k_per_cartoon=12` (top-12 vs. bottom-12 by composite score), build temperature 1.0, top-p 0.99, `build_gen_max_new_tokens=40`, `build_gen_batch_size=16`, `build_covo_batch_size=32`.
- **Data**: NYCCC, `top_percent=0.1`, cartoon-level `val_frac=0.1`, pair-level `pair_val_frac=0.1`, `n_contests_cap=null`.
- **Batching**: `per_device_train_batch_size=4`, `gradient_accumulation_steps=4` (effective batch 16).
- **Optimization (per iter)**: lr $=$ 5e$-$6, cosine schedule, warmup ratio 0.1, epochs $= 1$, bf16, `max_steps`$= -1$.
- **Sequence lengths**: `max_length=512`, `max_prompt_length=400`. `gen_eval_n_cartoons=20`, `policy_agreement_n_pairs=60`; decoded with `eval_temperature=` `eval_top_p=0.9`, `eval_max_new_tokens=40` (mirrors downstream eval; distinct from build-time temperature 1.0).
- **Eval / checkpointing**: `eval_steps=100` (TRL native pair eval), `logging_steps=10`, `save_steps=200`, `save_total_limit=2`.
- **Seed**: 42.

B

Prompts

Funny-word generation The prompt $\mathcal{P}_{\text{fun}}$ instructs the generator $\mathcal{G}$ to write a single word that replaces a word in the original headline at position z. After the replacement, the headline should be humorous.

```
You are a witty and humorous editor for a news publication.
    Your job is to make news headlines funnier by replacing
    a specific word with a more humorous alternative, while
    keeping the headline coherent and relevant to the
   original content. The replacement word has to be of the
   same part of speech. Respond only with the new word.
Headline: "<headline>". Task: Change word "<word>" (a <pos
   >) in the headline to make it funnier.
```

Serious-word generation The prompt $\mathcal{P}_{\text{serious}}$ instructs the generator $\mathcal{G}$ to write a single word that replaces the edit word $\hat{w}_z$ in the **edited** headline at position z. The replacement should revert the headline from humorous back to serious. This prompt is an auxiliary technique adopted from Franceschelli and Musolesi [2025] to compute the *value* term of the $CoVO$ score.

```
You are a serious journalist specialized in news headlines.
    Your job is to take silly headlines and make them
   serious by replacing the unfitting silly word
with a more serious and realistic alternative, keeping the
   headline coherent and relevant. The replacement word has
    to be of the same part of speech. Respond only with the
    new word.
Funny headline: "<headline>"
Task: Change word "<word>" (a <pos>) in the headline to
   make it a serious news headline.
```

Caption generation with cartoon description The prompt $\mathcal{P}_{\text{caption}}$ instructs the generator $\mathcal{G}$ to write a caption for a cartoon given its description and, in the case of a VL model, the cartoon image.

```
You are competing in The New Yorker Cartoon Caption Contest.
```

```
Write a short, funny caption for the cartoon. The caption
   should be something a character
in the cartoon might say, or a witty observation. Be
   concise -- one sentence only.
The cartoon's description is: <description>
Write a funny caption for this cartoon.
```

Caption generation with cartoon image and description

```
You are competing in The New Yorker Cartoon Caption Contest.

Write a short, funny caption for the cartoon. The caption
   should be something a character
in the cartoon might say, or a witty observation. Be
   concise -- one sentence only.
Here is the cartoon:
<image>
The cartoon's description is: <description>
Write a funny caption for this cartoon.
```

Caption description generation The prompt $\mathcal{P}_{\text{describe}}$ instructs the generator $\mathcal{G}$ to generate a description of a cartoon scene given its caption. This auxiliary prompt is used in the computation of the CoVO value term.

```
You are a cartoon analyst. Given a funny caption from The
   New Yorker Cartoon Caption Contest, describe the cartoon
    scene that this caption was written for. Be specific
   about the setting, characters, and entities.
Caption: "<caption>"
Describe the cartoon scene this caption was written for:
```

C

User study details

Participants received the following information upon beginning the survey:

> This study is conducted as part of Carl Lange's master's thesis "Generating humor using information-theoretic reward signals". A key challenge in computational humor research is evaluation: automated metrics correlate poorly with human judgment. This study collects direct human assessments to complement automated metrics. The task presents New Yorker-style cartoons alongside cartoon caption alternatives. Participants will rank caption alternatives based on their perceived funniness. No prior knowledge of AI or linguistics is required. The study is conducted entirely online and takes approximately 10–15 minutes to complete. The study involves minimal risk, and the chance of causing discomfort is not greater than in everyday life. This consent form will be stored at a safe location at Chalmers University of Technology.

The survey showed the following disclaimer:

> Humor is a sensitive and subjective topic, and individual reactions to humorous content may vary significantly. Although all captions included in this study have been manually reviewed to exclude overtly offensive material, we cannot guarantee that all participants will find every caption inoffensive. If at any point you feel uncomfortable with the content, you are free to withdraw from the study without consequence.

Participants were instructed about consent:

> By clicking "yes" below, you agree to, - participate voluntarily, - understand that you can terminate and/or recall participation at any time without consequences, - understand that the results generated by your participation will only be used for research and pedagogical training purposes. - understand that you, on request, can get access to the data you contributed as well as get information about how the data is utilized in the study. If you click "No" and continue in the survey, your data will not be processed.

Participants received instructions on how to complete the task:

In the following, you will be shown cartoon images. Each cartoon has a description and three caption alternatives. Your task is to rank the captions by how funny you find them. Put the funniest caption at the top, the least funny at the bottom. Some questions ask for an explanation of your choice. Write your answer in bullet points or sentences in the comment box. There are no right or wrong answers.