Table 7: Examples of Generated Cartoon Descriptions

Type of descriptions	GPT-4o	Human Written [20]
Canny description	A knight in armor is riding a horse, holding a lance with a traffic light on top. A line of businessmen in suits follows behind him.	There are two men on a horse. They are wearing soldier outfits. Businessmen follow behind them.
Uncanny Description	It's unusual to see a medieval knight leading modern businessmen as if going into battle.	There are businessmen following a two guys on horses who are soldiers.
Location	an open field	a hilly path
Entities	Knight, Horse, Businessmen, Traffic light	Warrior, Horses in warfare, Businessperson

A Links to Resources

Our dataset is available at https://huggingface.co/datasets/yguooo/newyorker_caption_ranking under Creative Commons Attribution Non Commercial 4.0. Our codebase is available at https://github.com/yguooo/cartoon-caption-generation under Apache 2.0.

B Language Model Prompts

B.1 Description Generation

We use GPT-4o to generate descriptions for each cartoon. In the dataset from Hessel et al. [20] each cartoon has a canny description, an uncanny description, a location, and a list of entity. Entity are words that is related to the cartoon. We used the five shot method to generate a set of descriptions. The five examples are randomly selected from the testing set, and we use the these same five example for every cartoon descriptions generation. An example of our prompt is shown below.

> **User**: In this task, you will see a cartoon, then write two descriptions about the cartoon, one uncanny description and one canny description, then write the cartoon's location, and the entities of the cartoon. I am going to give you five examples first and you write the last sets of description.
> **User**: <Insert Cartoon Image>
> **Assistant**: The canny description is <insert canny description> and the uncanny description is <insert uncanny description>, and the cartoon's location is <insert location>, and the entities of the cartoon are <insert entities>
> ......*Repeat user/assistant for four more examples*......
> **User**: <Insert Cartoon Image>. The set of description is

B.2 Caption Evaluation

We evaluate various models that generate captions by comparing the generated captions against four groups of human contestant entries at different ranking levels, which include top10, #200-#209, #1000-#1009, and contestant median. As concluded based on Table 2, we use GPT4-Turbo as evaluator with descriptions from Hessel et al. [20] in Overall Comparison and GPT4o-vision as evaluator with raw cartoon images in Best Pick Comparison. For both group comparison methods, we utilize the 5-shot in-context prompting technique, as mentioned in Section 4.2.

An example of Overall Comparison is shown below.

System: You are a judge for the new yorker cartoon caption contest.
User: In this task, you will see two description for a cartoon. Then, you will see two captions that were written about the cartoon. Then you will choose which captions is funnier. I am going to give you five examples first and you answer the last example with either A or B.
User: For example, the descriptions for the images are <Insert Canny Description> and <Insert Uncanny Description>. The two captions are A: <Insert CaptionA> B: <Insert CaptionB>
Assistant: The caption that is funnier is <Insert Answer>
......*Repeat user/assistant for four more examples*......
User: The descriptions for the images are <Insert Canny Description> and <Insert Uncanny Description>. The two groups of captions are group A: <Insert Caption Group A> group B: <Insert Caption Group B>
User: Choose the group of captions that is funnier. Answer with only one letter A or B, and nothing else.

An example of Best Pick Comparison is shown below.

System: You are a judge for the new yorker cartoon caption contest. Your job is to find the funniest caption.
User: In this task, you will see a cartoon first and two captions that were written about it then. The task is to choose which caption is funnier. I am going to show you five cartoons, corresponding captions and their answers first. In the end, for the last cartoon, answer with only one letter A or B, and nothing else.
User: <Insert Cartoon Image>
User: For this example, the two captions are A: <Insert CaptionA> B: <Insert CaptionB>. The answer is
Assistant: <Insert Answer>
......*Repeat user/assistant for four more examples*......
User: <Insert Cartoon Image>
User: Find the funniest caption for each group. Then choose the funnier group based on these funniest captions. Think step by step but finish the last line of your answer with only one letter A or B, and nothing else. A: <Insert Caption Group A> or B: <Insert Caption Group B>

B.3 Caption Generation

We used GPT-3.5-turbo, Claude-3-opus, and GPT-4-o to generate captions for each cartoons. We first use the system role to prompt it to generate 10 captions. Then we provide the image descriptions and then the image itself. For GPT-3.5-turbo, we simply only provided the image descriptions. For GPT-4-o, we have two versions where in one we provide the image itself, and the other we only provided the image descriptions. For Claude, we always provide both image description and image itself.

System: I want you to act as a sophisticated reader of The New Yorker Magazine. You are competing in The New Yorker Cartoon Caption Contest. Your task is to generate funny captions for a cartoon. Here are some ideas for developing funny captions. First think about characteristics associated with the objects and people featured in the cartoon. Then consider what are the unusual or absurd elements in the cartoon. It might help to imagine conversations between the characters. Then think about funny and non-obvious connections that can be made between the objects and characters. Try to come up with funny captions that fit the cartoon, but are not too direct. It may be funnier if the person reading the caption has to think a little bit to get the joke. Next, I will describe a cartoon image and then you should generate 10 funny captions for the cartoon along with an explanation for each.
User: <Insert Cartoon Image>
User: ”The cartoon’s description is: <insert canny description>.The uncanny description is: <insert uncanny description>. The location of the cartoon is:<insert location>. The entities of the cartoon are: <insert image entities>

C Additional Experiment Setups

C.1 Human Experiment Details

Each participant provided informed consent in compliance with our Institutional IRB and was compensated for their time. We paid participants $12 an hour and spent about $600 on data collection. The following instructions were used for the human experiments.

C.1.1 Human Pairwise with description generated by GPT4o-vision

In each trial of this task, you will see a description of a cartoon and two captions: the cartoon description is on the top, and the two caption choices are beneath the cartoon description. For each trial, please select the caption that is the funniest for the cartoon.

C.1.2 Human Pairwise with Cartoon Image

In each trial of this task, you will see one cartoon and two captions: the cartoon is on top, and the two caption choices are beneath the cartoon. For each trial, please select the caption that is the funniest for the cartoon.

C.1.3 Human Group (Overall) with description generated by GPT4o-vision

In each trial of this task, you will see a description of a cartoon and two groups of captions: the cartoon description is on the top, and the two grouped caption choices are beneath the cartoon description. For each trial, please select the group of captions that is the funniest for the cartoon.

C.1.4 Human Group (Overall) with Cartoon Image

In each trial of this task, you will see a cartoon and two groups of captions: the cartoon is on the top, and the two grouped caption choices are beneath the cartoon. For each trial, please select the group of captions that is the funniest for the cartoon.

C.1.5 Human Group (Best Pick) with description generated by GPT4o-vision

In each trial of this task, you will see a description of a cartoon and two groups of captions: the cartoon description is on the top, and the two grouped caption choices are beneath the cartoon description. For each trial, please select the group of captions that contains the funniest caption for the cartoon. First, pick the funniest caption in each group, and then compare between the two captions to pick the funniest group.

C.1.6 Human Group (Best Pick) with Cartoon Image

In each trial of this task, you will see a cartoon and two groups of captions: the cartoon is on the top, and the two grouped caption choices are beneath the cartoon. For each trial, please select the group of captions that contains the funniest caption for the cartoon. First, pick the funniest caption in each group, and then compare between the two captions to pick the funniest group.

C.1.7 Human top 10 vs Claude generated captions

In each trial of this task, you will see a cartoon and two groups of captions: the cartoon is on the top, and the two grouped caption choices are beneath the cartoon. For each trial, ¡strong¿please select the group of captions that is the funniest for the cartoon.

C.2 Recalibration of GPT Models for Ranking

For group comparisons without chain of thought, we observe a strong bias of GPT4 models choosing A over B. In other words, for some examples, the model always chooses option A even after we flip the two groups. Therefore, this suggests we need to calibrate the model predictions. We adopt a simple approach by readjusting the decision threshold. Let s_i^A, s_i^B denote the log probabilities of choosing A and B by the GPT4 model for two groups of human submitted captions x_i^A and x_i^B respectively. We use a small validation set of m examples $\{x_i^A, x_i^B\}_{i=1}^m$ with sigmoid scores $\{s_i^A, s_i^B\}_{i=1}^m$ and ground truth preference by the crowd denoted as $\{y_i \in \{A, B\}\}_{i=1}^m$. The current decision rule takes the form of $\widehat{f}(x_i^A, x_i^B) = \begin{cases} A & \text{if } s_i^A - s_i^B > 0 \\ B & \text{otherwise} \end{cases}$.

We simply set a different threshold τ, which induces $\widehat{f}_\tau(x_i^A, x_i^B) = \begin{cases} A & \text{if } s_i^A - s_i^B > \tau \\ B & \text{otherwise} \end{cases}$. The threshold τ^* is chosen so that the accuracy over the validation set is maximized:

$$\tau^* = \arg\max_\tau \sum_{i=1}^m \mathbf{1}\{y_i = \widehat{f}_\tau(x_i^A, x_i^B)\}.$$

Ties are broken arbitrarily above. We then use the recalibrated decision rule with τ^* for all of our evaluations.

C.3 Finetuning Experiment Details

Our training and test split for finetuning range from contest 530 to 890. In particular, our dataset includes all the data of [20] with ranking information within this range. ([20] only contains contests up to #763.) Thus, we choose our test split to be the combination of testing (47 contests) and validation split (44 contests) of [20] within the 530-890 range. The rest available contests form our training split.

Our finetuning methods are trained from Mistral 7B Instruct v0.1 and LLaVa v1.6 Mistral (multimodal case) via LoRA updates [21]. We use a variant of Mistral 7b model as our initial reward model to finetune from [3]. The choice of reward is based on our benchmarking results of top reward models on our caption generation dataset (Table 8). For SFT methods, we train on 1000 pairs of captions from each contest, with the preferred caption from the top 1000 captions and the alternative randomly sampled from the rest. For reward modeling, DPO and RLHF, we train on 1000 pairs of captions with three standard deviations apart according to Equation (1) per contest. Additionally, we train our model using the default choice of optimizer from TRL up to 1 epoch. Then, we search for the best hyper-parameter over the neighborhood of default parameters and pick the best performing model under our GPT-based group comparison metrics. For our reward model, we pick the best model based on the reward evaluation on the holdout set. For both pretrained and finetuned models, we use the same generation configuration file with temperature 0.7, top-p sampling probability 0.95, repetition penalty 1.15. When evaluating using the Best-of-N (BoN) method, we pick the top 10 captions based on the trained reward model, out of 50 generated candidates from caption generation models. Our choice of batch is 64 for SFT and reward model, and 128 for all other settings.

During the training process of DPO, PPO, SFT, we create a separate padding tokens and resize the token embedding of the pretrained model so that the text generation can terminate properly. Furthermore, in the loss design of SFT case, we only evaluate the next-token prediction loss on the caption segment, as all the training texts contain similar prompts. Since we only reported the iteration with the best results, early stopping occurs before a single epoch for the choice of best iterations.

Choice of Prompts In Table 10, we document the best prompt we found for each training algorithm. Generally speaking, the zero-shot, SFT, preference learning algorithm each require simpler prompts than the one preceding them.

Computation Cost Finetuning a SFT, DPO, PPO model usually takes 2-4 days to train till convergence on a A100 machine. Evaluating a single number of each scenario cost roughly \$5 on the openai platform.

[3]We use the pretrained reward model from `https://huggingface.co/weqweasdas/RM-Mistral-7B`

D Crowdsourced Caption Contest Ratings

Algorithm 1 Upper Confidence Bound (UCB) Algorithm

1: **Initialization:** For each caption x, initialize $N_x(0) = 0$ and $\hat{\mu}_x = 0$.
2: **for** $t = 1$ to T **do**
3: Select caption $x_t = \arg\max_x \left(\hat{\mu}_x + \sqrt{\frac{2\ln(4N_x(t)^2)}{N_x(t)}}\right)$.
4: Observe the reward $r_t \in \{1, 2, 3\}$ for caption x_t.
5: Update the number of times action x_t has been selected: $N_{x_t}(t) = N_{x_t}(t-1) + 1$.
6: Update the empirical mean reward of action x_t:

$$\hat{\mu}_{x_t} = \frac{N_{x_t}(t-1) \cdot \hat{\mu}_{x_t} + r_t}{N_{x_t}(t)}$$

7: **end for**

As described in the text, we used a UCB [2] variant to encourage high-performing captions to recieve the votes. We experimented with standard UCB (see Algorithm 1) and KL-UCB specifically optimized for discrete rewards [49]. The data repository labels datasets according to which algorithm was employed for each contest. In practice, using UCB in high-traffic asynchronous environments faces specific challenges. For example, we wanted to ensure that voters could only vote on one caption at a time, that the model sent batches of captions to users to reduce round trips to the server, and that the underlying model was able to update as frequently as possible. For more details on overcoming such challenges, see [23].

E Additional Results

We benchmark the performance of different reward model as in Table 8. weqweasdas/RM-Mistral-7B and Eurus-RM-7B Instruct are the top two models with the highest reward ranking accuracy. We choose to use weqweasdas/RM-Mistral-7B because it generally achieves better ranking accuracy for various data settings that we experimented on.

In our experiment, we noticed that PPO algorithm requires a much more aggressive early stopping scheme than DPO and SFT. Thus, we further look at the training dynamics of the PPO algorithm in Table 9. Here, the batch size is 128. It is worth noting that the result at iteration 0 has an lower overall win rate than the zero shot result in Table 3. The reason is that our PPO and DPO algorithms need to use a simpler prompt as in Table 10 to generate meaningful texts. From Table 9, we verified the steady increase of the mean reward and decrease of the training loss. However, the improvement on these metrics does not corresponds to an improvement of the overall humorous generation. We hypothesize that this is due to the complex nature of humor and the potential for out-of-distribution generations when running RLHF.

Table 8: Reward model benchmark

	Reward Ranking Acc (%)
Mistral-7B Instruct	73.17
Llama-3-8B Instruct	74.01
Llama-2-7B Chat	72.63
weqweasdas/RM-Mistral-7B	**74.05**
Eurus-RM-7B	**74.18**
FsfairX-LLaMA3-RM-v0.1	73.72
Qwen1.5-7B-Chat	72.26

Table 9: Training Dynamics of PPO

Iteration	0	10	20	30	40	50
Contestant Median (Overall Win Rate (%)) ↑	17.03	**24.73**	16.48	9.89	6.04	4.95
Mean Reward ↑	0.0057	0.0260	0.0186	0.1309	0.1356	**0.2587**
Loss ↓	0.3592	0.2001	0.1773	0.1709	0.0848	**0.0584**

Table 10: Best choice of prompts for each training algorthm

	Best Choice of Prompt
Zero-Shot	[INST] <> I want you to act as a sophisticated reader of The New Yorker Magazine. You are competing in The New Yorker Cartoon Caption Contest. Your task is to generate funny captions for a cartoon. Here are some ideas for developing funny captions. First think about characteristics associated with the objects and people featured in the cartoon. Then consider what are the unusual or absurd elements in the cartoon. It might help to imagine conversations between the characters. Then think about funny and non-obvious connections that can be made between the objects and characters. Try to come up with funny captions that fit the cartoon, but are not too direct. It may be funnier if the person reading the caption has to think a little bit to get the joke. Next, I will describe a cartoon image and then you should generate 1 funny caption for the cartoon along with an explanation for each. scene: <*scene*> description: <*description*> uncanny description: <*uncanny description*> entities: <*entities*> <> funny caption: [/INST] <*sample caption*>
SFT	[INST]I want you to act as a sophisticated reader of The New Yorker Magazine. You are competing in The New Yorker Cartoon Caption Contest. Your task is to generate funny captions for a cartoon. Here are some ideas for developing funny captions. First think about characteristics associated with the objects and people featured in the cartoon. Then consider what are the unusual or absurd elements in the cartoon. It might help to imagine conversations between the characters. Then think about funny and non-obvious connections that can be made between the objects and characters. Try to come up with funny captions that fit the cartoon, but are not too direct. It may be funnier if the person reading the caption has to think a little bit to get the joke. Next, I will describe a cartoon image and then you should generate 1 funny caption for the cartoon[/INST] scene: <*scene*> description: <*description*> uncanny description: <*uncanny description*> entities: <*entities*> funny caption: <*sample caption*>

	Best Choice of Prompt
LLaVA	[INST] I want you to act as a sophisticated reader of The New Yorker Magazine. You are competing in The New Yorker Cartoon Caption Contest. Your task is to generate funny captions for a cartoon. Here are some ideas for developing funny captions. First think about characteristics associated with the objects and people featured in the cartoon. Then consider what are the unusual or absurd elements in the cartoon. It might help to imagine conversations between the characters. Then think about funny and non-obvious connections that can be made between the objects and characters. Try to come up with funny captions that fit the cartoon, but are not too direct. It may be funnier if the person reading the caption has to think a little bit to get the joke. Next, I will provide a cartoon image with descriptions and then you should generate 1 funny caption for the cartoon along with an explanation for each. image: <*image*> scene: <*scene*> description: <*description*> uncanny description: <*uncanny description*> entities: <*entities*> Generate a funny caption for the image: [/INST] <*sample caption*>
DPO/PPO/Reward Model	scene: <*scene*> description: <*description*> uncanny description: <*uncanny description*> entities: <*entities*> funny caption: <*sample caption*>