Reasoning without Verifiers for Humor Generation

1 Introduction

SFT is insufficient for humor generation. In practice, supervised fine-tuning on humor datasets tends to improve surface-level fluency but it does not produce genuinely funny outputs. Models often converge to generic templates and safe puns, suggesting that pure likelihood training does not elicit creativity in humor generation. **References:**

- Loakman, Tyler and Thorne, William and Lin, Chenghua (2025). "Who's laughing now? an overview of computational humour generation and explanation."

Memorization is a symptom of SFT, RL may help generalization. A recurrent problem of SFT is memorization and imitation of frequent patterns from training data rather than generalization. Recent research in post-training methods suggests that RL objectives help generalize where SFT fails. **References:**

- Chu, Tianzhe et al. (2025). "Sft memorizes, rl generalizes: A comparative study of foundation model post-training."
- Gao, Leo et al. (2023). "Scaling laws for reward model overoptimization."

RLHF for humor is limited by unreliable reward and annotation noise. Standard RLHF relies on preference supervision and reward modeling. For humor, the underlying preference signal has high variance and low inter-annotator agreement, resulting in unstable training and poor final quality. **References:**

- Zhang, Jifan et al. (2024). "Humor in ai: Massive scale crowd-sourced preferences and benchmarks for cartoon captioning."

RLAIF for humor is unreliable because LLM judges do not match human humor understanding. Replacing human preferences with LLMs (RLAIF) is an great opportunity for scaling, but research shows model-based evaluation can systematically deviate from human preferences. As a result, the reward signal may become critically misaligned. **References:**

- Sakabe, Ritsu et al. (2025). "Assessing the Capabilities of LLMs in Humor: A Multi-dimensional Analysis of Oogiri Generation and Evaluation."

Agentic systems show promise for humor generation. In contrast to single-turn generation, multi-turn pipelines explicitly imitate creative practices. They generate associations, build narratives, draft and refine candidates. This allows them to yield better results, bridging the gap with humans in evaluation. **References:**

- Tikhonov, Alexey and Shtykovskiy, Pavel (2024). “Humor mechanics: Advancing humor generation with multistep reasoning.”
- Zhang, Jiajun et al. (2025). “HUMORCHAIN: Theory-Guided Multi-Stage Reasoning for Interpretable Multimodal Humor Generation.”
- Kim, Sean and Chilton, Lydia B (2025). “AI Humor Generation: Cognitive, Social and Creative Skills for Effective Humor.”

Agentic systems are architecturally restrictive. Multi-turn approaches effectively simulate creative reasoning, but they hard-code the reasoning structure (number of steps, branching strategy, theory choices). Different pipelines vary substantially, making it unclear which architectural decisions are fundamental and which are incidental. This motivates an alternative direction: rather than prescribing a reasoning procedure, attempt to train models to decide how to reason on their own.

DeepSeek-R1 suggests that RL can elicit diverse, unrestricted reasoning. Recent work on reasoning-oriented post-training shows that RL with verifiable rewards (RLVR) can elicit complex thinking behaviors without relying on human-authored CoT. This supports the idea that reasoning patterns do not have to be explicitly scripted in an agentic framework, they can emerge from the optimization objective. **References:**

- Guo, Daya et al. (2025). “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.”
- Lightman, Hunter et al. (2023). “Let’s verify step by step.”

Current reasoning models are biased towards verifiable domains. Empirically, many models refuse to sustain long reasoning when prompted with creative tasks, such as comedy writing. They often produce an early generic joke, most likely memorized from training data. When extended reasoning does occur through CoT prompting, it resembles human creative practices (consistent with the research on agentic systems): searching premises, exploring associations, iterating over formats, etc. Consequently, the outputs are often better, although the gap to remains large. A plausible explanation to the refusal is a training-induced bias: reasoning post-training is conducted in verifiable domains, such as math and code, so models learn a policy that does not transfer well to non-verifiable domains, such as humor. **References:**

- Wei, Jason et al. (2022). “Chain-of-thought prompting elicits reasoning in large language models.”

Models can learn to reason without verifiers. Recent work proposes alternatives that extend RLVR to domains without verifiers by using intrinsic model probabilities. Generally, non-verifiable tasks would require generating reasoning, an answer and running an arbitrary procedure (e.g., LLM-as-a-judge) to compare it with ground truth. Approaches like VeriFree and RLPR calculate rewards based on the likelihood of a reference given reasoning. It is a convenient approach from the computational standpoint because it .... Surprisingly, evaluation showed comparable or better results to RLVR checkpoints. These methods are candidates for adapting reasoning RL to humor generation, however there are a few limitations. **References:**

- Zhou, Xiangxin et al. (2025). “Reinforcing General Reasoning without Verifiers.”
- Yu, Tianyu et al. (2025). “RLPR: Extrapolating RLVR to General Domains without Verifiers.”

Reasoning in creative domains is limited by multiple and partial references. Creavity requires novelty and diversity by its very definition. In humor, there are multiple valid outputs for the same constraint, and any dataset contains only a partial subset of them. This violates the single-reference assumption used in VeriFree to prove mathematical equivalence to RLVR. The methodological contribution of this paper is to extend VeriFree to the multi-reference and partially observed case, with explicit analysis of the statistical consequences of such assumptions, and to test if this procedure can elicit reasoning for humor generation.

2 Methodology

2.1 Dataset collection

The idea is to convert jokes into prompts that represent weak constraints or seeds so that each prompt has multiple plausible joke references.

Sources. Large text joke dataset, e.g. Short Jokes ($\sim$270k) and rJokes ($\sim$550k). They offer broad topic coverage and many repeated keywords, making them ideal for constructing multi-reference prompts.

Prompt extraction. For each joke y, use a model (e.g, GPT, DeepSeek, Qwen) to generate $1-3$ prompts x that the joke can plausibly answer. Consider this joke from rJokes: “Atheism is a non-prophet organization”. The possible prompts are:

- “Write a pun using the word prophet”
- “Write a joke about atheism”

Model should be instructed to identify the premise, analyze the topic and structure of the joke, and choose relevant constraints (backtracking). This is important to avoid incoherent prompts, such as “write a joke about an organization”, that do not preserve the original comedic mechanism.

Inversion. After prompts are extracted from the corpus of jokes, group jokes by identical or near-identical prompts. The outcome is a dataset of pairs $(x,\ Y^*(x))$, where $Y^*(x) = \{y_1, \dots, y_n\}$ are jokes matching the seed x. This produces the key structural condition needed for multi-reference learning.

Distribution issues. The prompt distribution will likely be heavy-tailed: some prompts map to many jokes (popular topics and common mechanisms, e.g. puns), others to unique jokes (obscure ideas and rare joke structures). Rather than discarding one or the other, the plan is to use stratified sampling with temperature reweighting at training time.

2.2 Theoretical framework

2.2.1 Notation

Following the notation used in reasoning RL research, define:

- x: prompt (constraint).
- z: reasoning trace sampled by the policy (text within $< think >$ and $< /think >$ tags).
- y: answer sampled by the policy (text after $< /think >$ tag).

- y^*: reference answer.
- Policy factorization:

$$\pi_\theta(z, y \mid x) = \pi_\theta(z \mid x)\, \pi_\theta(y \mid x, z).$$

2.2.2 RLVR

Assume there is a single verifiable reference answer y^* per prompt x. Define the verifier reward:

$$R(y, y^*) = \mathbb{I}\{y = y^*\}.$$

The RLVR objective is:

$$J_{\text{RLVR}}(\theta \mid x, y^*) = \mathbb{E}_{z \sim \pi_\theta(z|x)}\, \mathbb{E}_{y \sim \pi_\theta(y|x,z)} \big[R(y, y^*)\big].$$

Gradient can be obtained directly or using REINFORCE, i.e. the following score-function identity:

$$\nabla_\theta \mathbb{E}_{u \sim p_\theta(u)}[f(u)] = \mathbb{E}_{u \sim p_\theta(u)} \big[f(u) \nabla_\theta \log p_\theta(u)\big].$$

Apply it to the joint distributino of (z, y):

$$\nabla_\theta J_{\text{RLVR}} = \mathbb{E}_{z,y} \Big[R(y, y^*)\ \nabla_\theta \log \pi_\theta(z, y \mid x)\Big].$$

Using the factorization:

$$\nabla_\theta \log \pi_\theta(z, y \mid x) = \nabla_\theta \log \pi_\theta(z \mid x) + \nabla_\theta \log \pi_\theta(y \mid x, z).$$

Hence the gradient becomes:

$$\nabla_\theta J_{\text{RLVR}} = \mathbb{E}_{z,y} \Big[R(y, y^*) \big(\nabla_\theta \log \pi_\theta(z \mid x) + \nabla_\theta \log \pi_\theta(y \mid x, z)\big)\Big].$$

References:

- Guo, Daya et al. (2025). “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.”

2.2.3 VeriFree

Assume there is a single reference answer y^* per prompt x. This time verifier is not available. Note that $\mathbb{E}_y[\mathbb{I}\{y = y^*\}] = P(y = y^*) = \pi_\theta(y^* \mid x, z)$. Hence, RLVR objective can be rewritten as:

$$J_{\text{RLVR}}(\theta \mid x, y^*) = \mathbb{E}_{z \sim \pi_\theta(z|x)} \big[\pi_\theta(y^* \mid x, z)\big]$$

Using the previous identity, define the verifier-free reward:

$$r_\theta(x, z, y^*) := \pi_\theta(y^* \mid x, z).$$

Then VeriFree objective is:

$$J_{\text{VF}}(\theta \mid x, y^*) := \mathbb{E}_{z \sim \pi_\theta(z|x)} \big[r_\theta(x, z, y^*)\big] = \mathbb{E}_{z \sim \pi_\theta(z|x)} \big[\pi_\theta(y^* \mid x, z)\big].$$

So $J_{\text{VF}} \equiv J_{\text{RLVR}}$ in scalar sense under the single-reference assumption. The key practical benefit: no decoding of y is required and calculation of $\pi_\theta(y^* \mid x, z)$ is parallelizable.

To explicitly obtain policy gradient of VeriFree, differentiate:

$$\nabla_\theta J_{\mathrm{VF}} = \nabla_\theta \sum_z \pi_\theta(z \mid x)\; r_\theta(x, z, y^*).$$

Product rule:

$$\nabla_\theta J_{\mathrm{VF}} = \sum_z \Big[\nabla_\theta \pi_\theta(z \mid x) r_\theta(x, z, y^*) + \pi_\theta(z \mid x) \nabla_\theta r_\theta(x, z, y^*)\Big].$$

Use $\nabla f = f \nabla \log f$:

$$\nabla_\theta J_{\mathrm{VF}} = \sum_z \pi_\theta(z \mid x) \Big[r_\theta(x, z, y^*) \nabla_\theta \log \pi_\theta(z \mid x) + \nabla_\theta r_\theta(x, z, y^*)\Big].$$

Now expand $\nabla_\theta r_\theta(x, z, y^*)$:

$$\nabla_\theta r_\theta(x, z, y^*) = r_\theta(x, z, y^*)\; \nabla_\theta \log r_\theta(x, z, y^*) = r_\theta(x, z, y^*)\; \nabla_\theta \log \pi_\theta(y^* \mid x, z).$$

Finally:

$$\nabla_\theta J_{\mathrm{VF}} = \mathbb{E}_{z \sim \pi_\theta(z|x)} \Big[r_\theta(x, z, y^*) \big(\nabla_\theta \log \pi_\theta(z \mid x) + \nabla_\theta \log \pi_\theta(y^* \mid x, z)\big)\Big].$$

2.2.4 Multi-reference VeriFree

For humor and creative domain in general, the single reference assumption fails. Let $Y^*(x) \subset \mathcal{Y}$ be the set of known good references for seed x. Define the multi-reference verified reward:

$$R(y, Y^*) = \mathbb{I}\{y \in Y^*\}.$$

Then RLVR objective becomes:

$$J_{\mathrm{MR}}(\theta \mid x, Y^*) = \mathbb{E}_z\; \mathbb{E}_{y \sim \pi_\theta(\cdot|x,z)} \big[\mathbb{I}\{y \in Y^*\}\big].$$

But:

$$\mathbb{E}_y \mathbb{I}\{y \in Y^*\} = P(y \in Y^* \mid x, z) = \sum_{y \in Y^*} \pi_\theta(y \mid x, z).$$

Then define the set version of verifier-free reward:

$$r_\theta(x, z, Y^*) := \sum_{y \in Y^*} \pi_\theta(y \mid x, z).$$

This quantity is a valid probability in $[0, 1]$. It is the probability of the event that the generated answer lands inside the reference set, or the union of correct completion events.

Therefore:

$$J_{\mathrm{MRVF}}(\theta \mid x, Y^*) = \mathbb{E}_{z \sim \pi_\theta(z|x)} \big[r_\theta(x, z, Y^*)\big].$$

Policy gradient is equivalent to VF up to reward formulation.

2.2.5 The Consequences of Incompleteness

Another problematic property of references in the creative domain is incompleteness. It is impossible to reliably exhaust all possibilities. It is plausible that there exist good jokes not present in the reference set. Also, computing $\sum_{y\in Y^*} \pi_\theta(y \mid x, z)$ may be expensive when $|Y^*|$ is large. A natural approximation is to sample a subset $Y \subset Y^*$ uniformly without replacement, and use:

$$r_\theta(x, z, Y) := \sum_{y\in Y} \pi_\theta(y \mid x, z).$$

To answer how good this estimate is, let each $y \in Y^*$ have inclusion probability $\rho = P(y \in Y) = \frac{|Y|}{|Y^*|}$. Then:

$$\mathbb{E}_Y[r_\theta(x, z, Y)] = \mathbb{E}_Y\Big[\sum_{y\in Y^*} \mathbb{I}\{y \in Y\}\pi_\theta(y \mid x, z)\Big] = \sum_{y\in Y^*} P(y \in Y)\pi_\theta(y \mid x, z) = \rho \sum_{y\in Y^*} \pi_\theta(y \mid x, z).$$

Finally:

$$r_\theta(x, z, Y) = \rho\, r_\theta(x, z, Y^*).$$

So the estimate is unbiased up to a constant scale ρ and simple reward standardization can eliminate bias. For example, Group-Relative Policy Optimization (GRPO) uses standardization to stabilize training. This fact will be useful later in training procedure formulartion.

... What about consistency and efficiency?

Another issue with this estimate is the gradient. Optimizing $r_\theta = \sum_{y\in Y^*} \pi_\theta(y \mid x, z)$ increases probability mass on observed references. But since $\sum_{y\in\mathcal{Y}} \pi_\theta(y \mid x, z) = 1$, increasing mass on Y^* necessarily decreases mass on $\mathcal{Y} \setminus Y^*$, which contains both bad jokes and unknown good jokes. This is the fundamental false-negative risk.

Kullback-Leibler Divergence (KL) can be used to reduce unwanted mass reallocation. Optimization problem with constrained optimization:

$$\max_\theta\ \mathbb{E}_{x,z}\big[r_\theta(x, z, Y^*(x))\big] \quad \text{s.t.} \quad \mathbb{E}_x\big[\mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))\big] \le \varepsilon.$$

The Lagrangian form:

$$\max_\theta\ \mathbb{E}_{x,z}\big[r_\theta(x, z, Y^*(x))\big]\ -\ \beta\, \mathbb{E}_x\big[\mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))\big].$$

This addition helps to preserve probability mass over unseen references during reallocation.

2.2.6 Training Procedure

For each prompt x, sample a group of G latents $z_1, \ldots, z_G \sim \pi_\theta(z \mid x)$. For each z_i, compute a multi-reference reward:

$$\hat{r}_i := r_\theta(x, z_i, Y^*(x)) = \sum_{y\in Y^*(x)} \pi_\theta(y \mid x, z_i).$$

Then update:

1. **Reasoning term.** Use a group RLOO baseline:

$$A_i = \hat{r}_i - \frac{1}{G-1}\sum_{j\neq i} \hat{r}_j$$

Whence:

$$g_z = \frac{1}{G-1}\sum_{i=1}^{G} A_i \nabla_\theta \log \pi_\theta(z_i \mid x).$$

This is the variance-reduction mechanism that is mathematically justified for the score-function term.

2. **Reference term.** Unlike the reasoning term, baselining here is not unbiased:

$$g_r = \frac{1}{G-1}\sum_{i=1}^{G} \hat{r}_i \nabla_\theta \log \pi_\theta(y \mid x, z_i).$$

3. **KL regularization:**

$$g_{\mathrm{KL}} = -\beta \nabla_\theta \mathrm{KL}(\pi_\theta(\cdot \mid x) \parallel \pi_{\mathrm{ref}}(\cdot \mid x)).$$

Overall policy update:

$$\nabla_\theta J_{\mathrm{MRVF}}(\theta \mid x, Y^*(x)) = g_z + g_r + g_{\mathrm{KL}}.$$

2.2.7 Note on Baselining

Baselines are unbiased in score-function estimators only when the baseline does not depend on the sampled variable whose log-probability gradient is taken.

- For the reasoning term, $z \sim \pi_\theta(z \mid x)$ is sampled. For any baseline $b(x)$ independent of z:

  $$\mathbb{E}_{z\sim\pi_\theta}\big[b(x)\nabla_\theta \log \pi_\theta(z \mid x)\big] = b(x)\nabla_\theta \sum_z \pi_\theta(z \mid x) = b(x)\nabla_\theta 1 = 0.$$

  So we can replace reward r_θ by advantage $A_\theta := (r_\theta - b(x))$ in the reasoning term term without bias.

- For the reference term, y is not sampled. The gradient $\nabla_\theta \log \pi_\theta(y^* \mid x, z)$ is not under an expectation over $y \sim \pi_\theta(\cdot \mid x, z)$. Therefore, there is no identity like $\mathbb{E}_y[\nabla \log \pi(y)] = 0$ to eliminate bias:

  $$\mathbb{E}_z\big[(r_\theta - b(x))\nabla \log \pi_\theta(y^* \mid x, z)\big] = \mathbb{E}_z\big[r_\theta \nabla \log \pi_\theta(y^* \mid x, z)\big] - b(x)\,\mathbb{E}_z\big[\nabla \log \pi_\theta(y^* \mid x, z)\big],$$

References:

- Zhou, Xiangxin et al. (2025). “Reinforcing General Reasoning without Verifiers.”

3 [WIP] Experiments

3.1 Baselines

The models used in experiments are from Qwen3 family, namely in 1.7B, 4B and 8B sizes and the following checkpoints:

1. Base
2. Instruct
3. Thinking
4. MRVF over Base

3.2 Evaluation plan

Humor evaluation is subjective and noisy. The plan is to evaluate over multiple axes, borrowing from Oogiri multi-dimensional annotation and rubric methods:

- **Constraint adherence.** Does this joke satisfy the constraint implied in the prompt?
- **Novelty.** Is this joke likely memorized?
- **Human-likeness.** Is this joke human-made or model-generated?
- **Funniness** Can this joke make people laugh?

Axes should be calibrated rubric scores from 1 to 5, not scalar ratings, i.e. each level should be verbalized and distinct.

Two complementary evaluation strategies: small-scale human annotation and full-scale LLM-as-a-judge annotation. This is motivated by the evidence that LLMs and humans can critically diverge in humor judgement.

References:

- Hashemi, Helia et al. (2024). "LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts."
- Gunjal, Anisha et al. (2025). "Rubrics as rewards: Reinforcement learning beyond verifiable domains."
- Sakabe, Ritsu et al. (2025). "Assessing the Capabilities of LLMs in Humor: A Multi-dimensional Analysis of Oogiri Generation and Evaluation."

3.3 Ablations

- **Reference subset size $|Y^*|$.** How sensitive is training to incomplete reference observation? Is the scale-invariance and rationale valid?
- **KL hyperparameter β.** Are catastrophic forgetting or mode seeking detectable?
- **...**