Reasoning without Verifiers for Humor Generation

Me

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 Multidimensional 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. In humor, there are multiple valid outputs for the same constraint, and any dataset contains only a partial subset of them. Creavity requires novelty and diversity by its very definition. This violates the single-reference assumption used in VeriFree to prove mathematical equivalence to RLVR. The methodological contribution of this thesis is to extend VeriFree to the **multi-reference, partially observed** regime, with explicit attention to the statistical consequences of incomplete reference sets, and to use the resulting method to elicit creative reasoning behaviors for humor generation.

2 Methodology

2.1 Dataset collection: "joke → seed(s) → multi-reference prompt"

The data plan is to invert jokes into prompts that represent *weak constraints* ("seeds") so that each prompt has **multiple** plausible joke references.

Raw sources. Large text joke datasets (e.g., Short Jokes / rJokes) provide breadth and repeated themes (useful for multi-reference prompts). (Dataset sourcing details will be finalized; the key methodological point is prompt inversion.)

Seed extraction (LLM-assisted). For each joke y, generate $1-3$ seeds x that the joke plausibly answers. The intended seeds are *imperative prompt-like constraints*, e.g.:

- "Make a pun using *prophet* and *organization*"
- "Write a one-liner about atheism using a wordplay twist"

and **avoid** incoherent inversions ("write a joke about an organization") that do not preserve the original comedic mechanism.

This step is inspired by work showing LLMs can manipulate humor via controlled edits ("unfunning"), suggesting they can reliably localize the "comedic mechanism" even if they cannot always generate strong humor from scratch.

Inversion. After seeds are extracted from many jokes, group jokes by identical/near-identical seeds. The outcome is a dataset of pairs:

$$(x,\ Y^*(x)) \quad \text{where } Y^*(x) = \{y_1, \ldots, y_M\} \text{ are jokes matching seed } x.$$

This produces the key structural condition needed for multi-reference learning.

Practical distribution issues. The seed distribution will be heavy-tailed: some prompts map to many jokes (common themes), others to unique jokes (rare mechanisms). Rather than discarding one regime, the plan is to use stratified sampling / temperature reweighting at training time.

References:

- Horvitz, Zachary et al. (2024). "Getting serious about humor: Crafting humor datasets with unfunny large language models."

2.2 Formal framework and notation (preliminaries)

2.2.1 Policy with latent "reasoning" variable

Following the notation used in verifier-free reasoning RL discussions, define:

- x: prompt/seed (constraint).
- z: latent "reasoning" trace / plan / hidden variable sampled by the policy (can be instantiated as a reasoning text prefix, a plan, or a tool-free "scratch" segment).
- y: final answer text (joke).
- Policy factorization:

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

This decomposition isolates "how the model thinks" (z) from "what it outputs" (y). It is also the form used in RLVR-style derivations.

References:

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

2.3 RLVR: objective and REINFORCE gradient (step-by-step)

2.3.1 Single-reference RLVR objective

Assume (for now) **one** reference answer y^* per prompt x. Define the verifier reward:

$$R(y, y^*) = \mathbf{1}\{y = y^*\}.$$

The RLVR objective is:

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

Because $\mathbb{E}_y[\mathbf{1}\{y = y^*\}] = \Pr(y = y^*) = \pi_\theta(y^* \mid x, z)$,

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

This identity is the bridge that later enables verifier-free variants: the "sparse 0/1 reward after sampling y" equals the *probability mass* assigned to the reference under the policy.

2.3.2 REINFORCE gradient for RLVR

Start from the 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 sampling of (z, y):

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

Using the factorization,

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

So the REINFORCE gradient becomes:

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

This matches the structure in the notes and is the baseline for comparing VeriFree-style estimators.

References:

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

2.4 VeriFree: objective equivalence and gradient decomposition

2.4.1 VeriFree objective (single reference)

Using the identity $J_{\mathrm{RLVR}} = \mathbb{E}_z[\pi_\theta(y^* \mid x, z)]$, define the *verifier-free reward*:

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

Then the VeriFree objective is:

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

So $J_{\mathrm{VF}} \equiv J_{\mathrm{RLVR}}$ as scalar objectives under the single-reference assumption.

2.4.2 Gradient of VeriFree (explicit derivation)

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 $\nabla\pi = \pi\nabla \log \pi$:

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

Now expand $\nabla_\theta r_\theta$. Since $r_\theta = \pi_\theta(y^* \mid x, z)$,

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

Therefore:

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

This is the form in the notes, and it is the key practical benefit: **no sampling of** y is required (removing extreme sparsity), yet the objective remains equivalent in the single-reference case.

2.4.3 Why "baselining is only valid for the first term" (precise statement)

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 z-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 r_θ by $(r_\theta - b(x))$ in the z-**score** term without bias:

$$\mathbb{E}_z\big[r_\theta \nabla \log \pi(z)\big] \equiv \mathbb{E}_z\big[(r_\theta - b(x))\nabla \log \pi(z)\big].$$

- For the y^*-term in VeriFree, **there is no sampling of** y. The gradient $\nabla_\theta \log \pi_\theta(y^* \mid x, z)$ is not under an expectation over $y \sim \pi_\theta(\cdot \mid x, z)$; it is evaluated at a *fixed string* y^*. Therefore, there is no identity like $\mathbb{E}_y[\nabla \log \pi(y)] = 0$ to kill a baseline term. Concretely, if one attempted to baseline this term:

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

  and the final expectation is **not zero in general**, so the baseline would bias the gradient.

This is the clean reason the notes label RLOO/baselining as naturally applicable to the "reasoning sampling" part (the score term for z), but not automatically to the "reference-logprob" part.

References:

- Zhou, Xiangxin et al. (2025). "Reinforcing General Reasoning without Verifiers."
- Yu, Tianyu et al. (2025). "RLPR: Extrapolating RLVR to General Domains without Verifiers."
- Lanchantin, Jack et al. (2025). "Bridging Offline and Online Reinforcement Learning for LLMs."

2.5 Multi-reference VeriFree (MRVF): objective, reward, and key properties

2.5.1 Multi-reference objective

For humor, 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:

$$R_{\text{set}}(y, Y^*) = \mathbf{1}\{y \in Y^*\}.$$

Then the RLVR-style objective becomes:

$$J_{\text{MR}}(\theta \mid x, Y^*) = \mathbb{E}_z\, \mathbb{E}_{y\sim\pi_\theta(\cdot|x,z)}\big[\mathbf{1}\{y \in Y^*\}\big] = \mathbb{E}_z\big[\Pr(y \in Y^* \mid x, z)\big].$$

But

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

So define the **probability-mass reward** ("set mass reward"):

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

This quantity is a valid probability in $[0, 1]$: it is the probability that a sampled completion lands inside the reference set. It is not "a full distribution," but it does not need to be—it's an **event probability** under $\pi_\theta(\cdot \mid x, z)$.

Therefore:

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

2.5.2 Why $\sum$ is principled, but max is problematic

- **Sum** corresponds exactly to the probability of the event $y \in Y^*$. It uses *all* references and provides smooth gradients:

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

- **Max** reward $r_\theta^{\max} = \max_{y \in Y^*} \pi_\theta(y \mid x, z)$ is *not* the event probability. It is a lower-dimensional proxy that is (i) non-smooth at ties, and (ii) mode-seeking: it encourages the policy to put all mass on the single easiest-to-increase reference and ignore the rest. This is especially undesirable in humor, where diversity and mechanism coverage matter.

References:

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

2.6 Random reference subsampling and scale-invariance (explicit expectation proof)

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 $S \subset Y^*$ uniformly without replacement, $|S| = m$, and use:

$$\hat{r}_\theta(x, z; S) := \sum_{y \in S} \pi_\theta(y \mid x, z).$$

Expectation. Let each $y \in Y^*$ have inclusion probability $\rho = \Pr(y \in S) = \frac{m}{|Y^*|}$. Then:

$$\mathbb{E}_S[\hat{r}_\theta(x, z; S)] = \mathbb{E}_S\Big[\sum_{y \in Y^*} \mathbf{1}\{y \in S\}\pi_\theta(y \mid x, z)\Big] = \sum_{y \in Y^*} \Pr(y \in S)\pi_\theta(y \mid x, z) = \rho \sum_{y \in Y^*} \pi_\theta(y \mid x, z) = \rho\, r_\theta^{\mathrm{MR}}(x, z;$$

So the estimator is unbiased **up to a constant scale** ρ.

Why scale can be "free" under GRPO-style normalization. GRPO implementations commonly normalize rewards/advantages within a prompt-group (e.g., subtract mean, divide by std across samples). If all rewards are multiplied by a positive constant ρ, then (mean-centered, variance-normalized) advantages are unchanged. This motivates treating ρ as irrelevant for the

policy-gradient direction when using group-normalized advantages, which is why "random selection" can be plugged in without needing exact $|Y^*|$ in the inner loop—provided the optimizer indeed uses such normalization.

References:

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

2.7 The positive–unlabeled (PU) problem and why KL is not optional

2.7.1 Where the bias comes from

In open-ended domains, Y^* is incomplete: there exist good jokes not present in the reference set. Optimizing $r_\theta^{\mathrm{MR}} = \sum_{y \in Y^*} \pi_\theta(y \mid x, z)$ 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.

2.7.2 KL-constrained optimization (formal statement)

To reduce destructive mass reallocation, use constrained optimization:

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

The Lagrangian form:

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

This makes the "don't destroy the base distribution" prior explicit. It also aligns with mainstream RLHF practice and with the observation that RL generalizes but can overoptimize noisy reward models; KL helps control that regime.

References:

- Gao, Leo et al. (2023). "Scaling laws for reward model overoptimization."
- Chu, Tianzhe et al. (2025). "Sft memorizes, rl generalizes: A comparative study of foundation model post-training."
- Lanchantin, Jack et al. (2025). "Bridging Offline and Online Reinforcement Learning for LLMs."

2.8 Training plan: GRPO + RLOO-style baselines for z, plus reference-logprob terms

2.8.1 What is actually optimized (proposed final objective)

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 (possibly subsampled) multi-reference reward:

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

Then update:

1. **z-policy (search over reasoning/plans):** use a group baseline (RLOO-like)

$$A_i := \hat{r}_i - \frac{1}{G-1}\sum_{j\neq i}\hat{r}_j, \qquad g_z \approx \sum_{i=1}^{G} A_i \nabla_\theta \log \pi_\theta(z_i \mid x).$$

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

2. **Reference-probability term (non-score "pathwise" derivative of probabilities):** add the gradient of $\hat{r}_i$ w.r.t. θ. Using the log-derivative trick at the reference level:

$$\nabla_\theta \pi_\theta(y \mid x, z_i) = \pi_\theta(y \mid x, z_i)\ \nabla_\theta \log \pi_\theta(y \mid x, z_i),$$

   so

$$g_{\text{ref}} \approx \sum_{i=1}^{G}\sum_{y\in S_i} \pi_\theta(y \mid x, z_i)\ \nabla_\theta \log \pi_\theta(y \mid x, z_i).$$

   (Unlike the z-term, baselining here is not automatically unbiased.)

3. **KL regularization:**

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

Overall update direction:

$$g \approx g_z + g_{\text{ref}} + g_{\text{KL}}.$$

This matches the conceptual structure in the notes:

- "group mean / leave-one-out" applies where sampling happens (z),
- the probability-mass reward supplies smoother gradients than sparse acceptance,
- KL is the mechanism to mitigate PU collapse.

2.8.2 Why this is not "just SFT"

It is tempting to say "if references are given, maximizing their probability is SFT." The difference is *where the stochasticity and credit assignment live*:

- **SFT** trains on fixed (x, y) pairs and pushes up likelihood of each observed reference independently (often averaged across dataset). It does not explicitly create a *competition* among alternative reasoning paths z for the same x, and it does not use on-policy sampling to search for high-reward internal plans.
- **MRVF-style training** treats the model as a latent-variable policy: it samples multiple z per x, measures which z makes the reference set more probable, and shifts probability toward those z. This is closer to "test-time search distilled into policy improvement," and is conceptually aligned with why RL can generalize where SFT tends to memorize.

References:

- Chu, Tianzhe et al. (2025). "Sft memorizes, rl generalizes: A comparative study of foundation model post-training."
- Guo, Daya et al. (2025). "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning."
- Zhou, Xiangxin et al. (2025). "Reinforcing General Reasoning without Verifiers."

3 Experiments (planned)

3.1 Baselines

1. **Prompt-only:** direct constrained humor prompting (no latent z, no multi-step).
2. **Multi-step prompting / agent pipeline:** Humor Mechanics-like association $\rightarrow$ drafting $\rightarrow$ selection.
3. **SFT baseline** on (x, y) derived from seed inversion (single-reference and multi-reference variants).
4. **Preference-style tuning** if available: use pairwise comparisons (e.g., for captioning-like tasks there exist massive preference datasets, but for pure text jokes this may require building a small preference set).
5. **MRVF training** (this proposal): GRPO/RLOO on z + probability-mass reward + KL.

References:

- Tikhonov, Alexey and Shtykovskiy, Pavel (2024). "Humor mechanics: Advancing humor generation with multistep reasoning."
- Zhang, Jifan et al. (2024). "Humor in ai: Massive scale crowd-sourced preferences and benchmarks for cartoon captioning."

3.2 Evaluation plan

Humor is subjective and noisy; the plan is to evaluate along **multiple axes**, borrowing from Oogiri-style multi-dimensional annotation and rubric methods:

- **Constraint adherence** (hard checks where possible: required words/themes/format).
- **Novelty / non-memorization:** overlap-based heuristics + nearest-neighbor retrieval against training jokes.
- **Human-likeness** (does it read like a human joke vs. "LLM joke").
- **Funniness** (human pairwise or small-scale Likert).

Two complementary evaluation strategies:

- **Small human study:** pairwise preferences among a small set of methods for a curated set of seeds (lower sample size but reliable).
- **Rubric-based LLM judging with calibration:** use LLM-Rubric-style calibrated rubric scoring, not a single "rate funniness 1–5" prompt.

This is motivated by evidence that raw LLM judgments and human judgments can diverge on humor dimensions; calibrating may reduce noise.

Additionally, the "unfunning" transformation can be used for **counterfactual controls**: if a method produces "highly funny" outputs, a strong humor editor should be able to systematically remove the humor while preserving semantics, enabling downstream checks that the humor is not purely superficial word salad.

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."
- Horvitz, Zachary et al. (2024). "Getting serious about humor: Crafting humor datasets with unfunny large language models."
- Sakabe, Ritsu et al. (2025). "Assessing the Capabilities of LLMs in Humor: A Multidimensional Analysis of Oogiri Generation and Evaluation."

3.3 Primary ablations (to test the central hypothesis)

- **Effect of z-search (group size G):** does increasing reasoning-path samples improve final humor metrics?
- **Reference subset size m:** how sensitive is training to partial reference observation? (tests the scale-invariance rationale.)
- **KL strength β:** detect PU collapse / mode seeking vs. under-training.
- **Mass reward vs. max reward:** verify empirically that max collapses to a single reference mode and harms diversity.

References:

- Yu, Tianyu et al. (2025). "RLPR: Extrapolating RLVR to General Domains without Verifiers."
- Zhou, Xiangxin et al. (2025). "Reinforcing General Reasoning without Verifiers."
- Guo, Daya et al. (2025). "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning."