Corrective Feedback-Driven Humor Generation for Tiny Language Models

Ramakrishna Pinninti[1]*, **Dibyanayan Bandyopadhyay**[2,*], **Lalit Chandra Routhu**[2]
Gaël Dias[3], **Alexis Lechervy**[3], **Jeremie Pantin**[3]
Mohammed Hasanuzzaman[1], **Asif Ekbal**[2]
[1]Munster Technological University, Cork, Ireland
[2]Indian Institute of Technology Patna, Patna, India
[3]University of Caen Normandy, Caen, France
Correspondence: ramakrishna.pinninti@mymtu.ie

Abstract

Humor generation remains a challenging problem for language models due to its subjective, creative, and culturally sensitive nature. While large language models (LLMs) demonstrate strong generative capabilities, their deployment is often constrained by computational cost. We propose a lightweight, corrective feedback-driven framework in which a compact generator (GPT-2, 140M parameters) produces punchlines under the guidance of an LLM acting as a critic. Unlike traditional knowledge distillation, our method leverages the LLM as a feedback provider rather than a teacher to be mimicked, enabling the student model to iteratively refine its outputs through corrective supervision. We systematically evaluate multiple LLMs (LLaMA3-8B, Qwen3-8B and Mistral-7B) as guides under different strategies, including frozen and learnable configurations, to study their impact on humor quality and diversity. To support responsible humor generation, we curate a high-quality setup–punchline dataset from Reddit and explicitly filter out harmful content. Experimental results show that corrective feedback-guided training substantially improves humor quality, fluency, and diversity over standard fine-tuning and knowledge distillation baselines, while preserving the efficiency of compact models. We observe that, among the evaluated guides, Mistral-7B consistently provides particularly strong guidance on the curated joke dataset. Our code and resources are available at: https://github.com/Ramakrishna176420/Corrective-Feedback-Driven-Humor-Generation-for-Tiny-Language-Models.

1 Introduction

Humor plays an important role in human communication and engagement. Generating humor computationally, particularly in the setup–punchline format, remains a challenging problem in natural language generation. Despite progress in humor detection (Barral et al., 2017; Ziser et al., 2020; Arora et al., 2022) and conversational humor systems (Mundhra et al., 2021), producing high-quality jokes remains difficult due to humor's subjectivity, cultural dependency, and subtle linguistic structure.

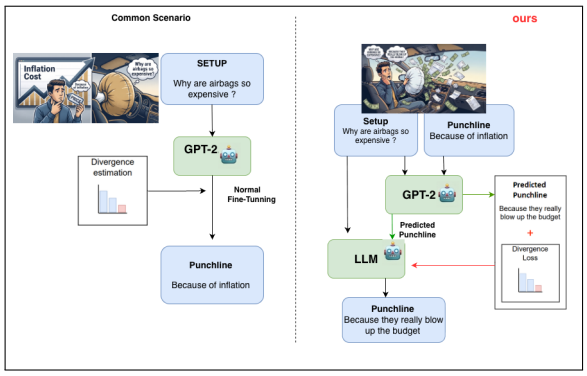


Figure 1: Comparison of state-of-the-art (SOTA) approaches with the proposed corrective feedback-driven humor generation framework.

Early computational humor systems relied on rule-based or template-based methods (Binsted and Ritchie, 1994), which lacked linguistic flexibility and creativity (Amin and Burghardt, 2020). Neural approaches, including LSTM and sequence-to-sequence architectures, improved fluency but still struggled to produce genuinely humorous content (Yu et al., 2018; Chen et al., 2024b). Transformer-based models such as GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020) and T5 (Raffel et al., 2020) further improved text generation, yet humor generation remains inconsistent (Li et al., 2023). Recent large language models (LLMs), including Mistral[1], LLaMA (Touvron et al., 2023), and Qwen (Yang et al., 2025), have significantly advanced text generation capabilities. However, humor generation remains difficult because it requires reasoning about incongruity, cultural context, and linguistic

*Equal contribution.

[1]https://huggingface.co/mistralai/Mistral-7B-v0.1

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 2898–2916
November 6-10, 2026 ©2026 Association for Computational Linguistics

creativity (Loakman et al., 2025; Zangari et al., 2025). For instance, recent surveys highlight that current AI systems still fall short of human-level humor understanding and generation (Lemmens and De Marez, 2026; Loakman et al., 2025). Empirical studies further show that LLMs often recycle familiar joke templates rather than producing original punchlines (Jentzsch and Kersting, 2023), and frequently rely on superficial lexical cues when interpreting humor (Zangari et al., 2025). These findings suggest that generating creative and contextually appropriate humor remains an open challenge.

Humor generation approaches that rely on LLMs require substantial computational resources. But, in real-world applications such as conversational agents and mobile assistants, strict constraints on latency, memory, and energy consumption often make large models impractical. As a consequence, knowledge distillation (KD) has recently been explored for humor generation (Ravi et al., 2024), where smaller student models imitate larger teacher models. However, imitation-based KD often encourages surface-level replication rather than creative reasoning, limiting the originality and diversity of generated jokes.

To address these limitations, we propose a framework where a compact generator model (e.g., GPT-2) produces punchlines while being guided by a large language model (e.g. Mistral-7B). Instead of imitating the teacher's outputs, the generator receives corrective feedback from the LLM, enabling it to refine its outputs while preserving efficiency. As illustrated in Figure 1, this approach allows compact models to learn humor-relevant patterns without directly mimicking large models.

We evaluate our approach using Reddit-based setup–punchline datasets (Weller and Seppi, 2020; Pungas, 2017). In particular, to promote responsible humor generation, we construct a high-quality setup–punchline dataset carefully removing any harmful content. Experiments comparing our framework with strong baselines, including fine-tuned models and KD-based methods, demonstrate that our approach achieves high humor quality and great diversity, as evidenced by automatic metrics, LLM-as-judge comparisons, and human annotations. Our work addresses the following research questions: **R1:** Can a compact student model guided by a LLM balance humor quality and diversity? **R2:** Can such a framework provide better efficiency while maintaining competitive humor quality compared to fine-tuning and KD methods? The main contributions of our paper are listed as follows:

1. We propose a corrective feedback-driven framework for humor generation using a compact generator guided by a LLM,

2. We show that our method improves funniness compared to standard fine-tuning and KD approaches,

3. We introduce a metric for measuring diversity in humor generation,

4. We demonstrate that compact models guided by LLM feedback can approach the humor quality of larger models while remaining computationally efficient,

5. We curate a setup–punchline humor dataset designed to support the generation of non-harmful jokes.

2 Related Work

Humor Generation. The field of humor generation has progressed from early template-based systems, such as JAPE (Binsted and Ritchie, 1994), which relied on fixed rules but often lacked originality (Amin and Burghardt, 2020). Neural methods marked a shift, beginning with LSTM-based model, though it struggled to produce consistently humorous text. Sequence-to-sequence (Seq2Seq) models further advanced the field: Yu et al. (2018) trained a conditional neural language model specifically for punning jokes, enhancing outputs using a specialized decoding algorithm.

LLM-guided Strategies. With the rise of LLMs, structured reasoning has been explored. Wang et al. (2025) introduced the CloST framework, refining humor with associative instruction tuning. Tikhonov and Shtykovskiy (2024) applied multi-step reasoning to reconstruct joke structures, while Zhong et al. (2024) proposed Leap-of-Thought for improved humor diversity. Recent studies also analyzed the humor generation capabilities of modern LLMs in conversational contexts. In particular, Quan et al. (2025) show that although LLMs can generate fluent and stylistically varied humor, they often struggle with contextual nuance and role interpretation, highlighting limitations in pragmatic humor understanding.

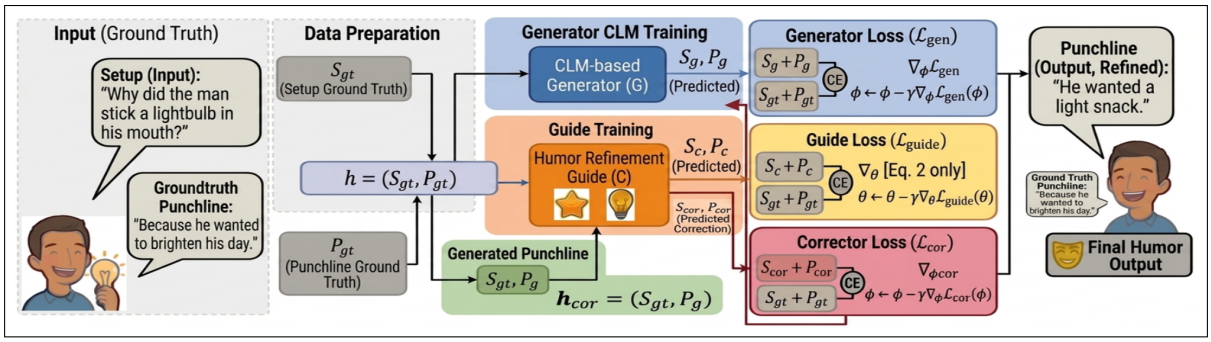


Figure 2: Our framework. S_{gt}: Ground truth setup, P_{gt}: Ground truth punchline, which together constitutes the humorous input h. P_g: Generated punchline from the S_{gt} by G. h_{cor} is obtained by concatenating S_{gt} and P_g. S_c, P_c are the setup-punchline pairs when h is given as an input to C_θ. S_{cor}, P_{cor} are the setup-punchline pairs when h_{cor} is given as an input to C_θ. CE stands for the cross-entropy loss.

Learning to Generate Humor. Fine-tuning methods, including parameter-efficient fine-tuning and RankGen (Krishna et al., 2022), have been applied to LLMs for improved joke coherence and humor detection. However, ChatGPT has been observed to recycle existing jokes, highlighting ongoing challenges in humor originality and contextual adaptability (Jentzsch and Kersting, 2023). Mirowski et al. (2024) examined LLMs as comedic support tools, finding that while LLMs assist in joke structuring, they lack cultural sensitivity, comedic timing, and deeper contextual understanding.

Lightweight Humor Generation. Chen et al. (2024a) proposed FrugalGPT, which learns which combinations of LLMs to use for different queries in order to reduce cost and improve accuracy. Their experiments show that FrugalGPT can match the performance of the best individual LLM (e.g. GPT-4) with up to 98% cost reduction. Ravi et al. (2024) explored the use of feedback in distilling knowledge from LLMs to Small Language Models (SLMs). By assigning LLMs a dual role (*teacher* and *critique*), their study demonstrates that incorporating feedback improves the performance of SLMs over traditional imitation-based distillation.

Knowledge Distillation for Language Models. (Kim and Rush, 2016) proposed Sequence-Level Knowledge Distillation, which extends conventional token-level distillation by training the student to imitate complete sequences generated by a teacher model. Gu et al. (2024) introduced MiniLLM, a distribution-level knowledge distillation framework that transfers knowledge from a larger teacher model to a smaller student by minimizing KL divergence between their output distributions. This enables the student to better align its generation behavior with the teacher while retaining efficient inference. Thus, while MiniLLM transfers knowledge through teacher-student distribution alignment, our framework uses the LLM as a corrective guide for humor generation rather than as a distribution to imitate. Agarwal et al. (2024) introduced On-Policy Distillation, where the student learns from its own generated outputs under teacher supervision to reduce train–test mismatch. While this is similar to our use of student-generated punchlines, On-Policy Distillation aligns the student with the teacher distribution, whereas our method uses the LLM as a corrective guide without requiring direct imitation. Lin et al. (2020) proposed Autoregressive Knowledge Distillation through Imitation Learning, which trains the student on its own autoregressive generations while receiving token-level or distribution-level supervision from the teacher. The approach addresses exposure bias by incorporating student-generated trajectories during training. While this method relies on teacher-policy imitation, our approach uses the LLM to provide corrective feedback on the student-generated punchline rather than directly imitating the teacher's token decisions or output distribution. Ko et al. (2024) introduced DistiLLM, a knowledge distillation framework designed for autoregressive language models. The method addresses two limitations of conventional distillation approaches: the lack of a standardized objective for autoregressive sequence models and the additional computational cost introduced when student-generated outputs are incorporated during training. DistiLLM combines a skew KL divergence objective with an adaptive off-policy strategy, allowing the student to learn from the teacher distribution while making more

efficient use of its own generated outputs. By incorporating student-generated samples, the method is intended to reduce the mismatch between training and inference that arises when the student must generate autoregressively at test time. The adaptive off-policy component further improves efficiency by controlling how student-generated data are used during optimization. While DistiLLM remains a distribution-level distillation approach that aligns the student with the teacher through KL-based objectives, our method instead uses the LLM as a corrective guide that provides supervision on the student-generated punchline without directly matching the teacher distribution. While these approaches primarily transfer knowledge through teacher-sequence imitation, distribution matching, or teacher-policy supervision, our framework instead uses the LLM as a corrective guide that provides feedback on the student-generated punchline without requiring direct imitation of the guide's outputs.

3 Datasets

To construct our setup–punchline dataset, we combine two Reddit-based datasets (Weller and Seppi, 2020; Pungas, 2017). The data consists of titles and bodies with upvote counts, where titles serve as setups and bodies as punchlines. We filter entries by length to preserve the setup–punchline format and split the data into training and test sets using a 80–20 ratio. To ensure training quality, only the top 20,000 jokes ranked by upvotes are used for training. Instead, the test set contains 30,433 jokes without upvote constraints to evaluate generalization capabilities across diverse learning setups.

Curated vs. Non-Curated Data. We report results on two variants of our dataset. The *non-curated* split (UF) includes structurally filtered jokes but may contain harmful content such as stereotypes, bullying, hateful conduct, or profanity. Instead, the *curated* split (F) is obtained by automatically filtering such content using a LLM-based safety classifier following the original ideas of Vikhorev et al. (2024) and Lu et al. (2025). For each setup–punchline pair, a Qwen3-8B model is prompted with a definition of offensive language and asked to respond with "yes" or "no", whether the joke is harmful or not. Jokes labeled offensive are removed. Filtering is applied to both training and test sets. Note that results on UF assess generalizability, but are not used for deployment or released to avoid harmful content generation.

4 Methodology

As shown in Figure 2, our framework consists of two generative models: a generator G_ϕ (a compact model like GPT-2), which generates a punchline for a given setup, and a guide C_θ (a LLM like Mistral-7B), which critics and refines the jokes[2].

At the training phase, G_ϕ learns to generate better punchlines, guided by C_θ that corrects the initial errors made by G_ϕ. We use the guide in a frozen state with 4-bit quantization, enabling efficient feedback without the computational overhead of training LLMs. We also explore variants where the guiding model is learnable.

Unlike critique-based distillation approaches such as Ravi et al. (2024), our framework does not require the student model to reproduce the teacher outputs. The generator first produces its own candidate punchlines, and the guiding LLM then provides corrective supervision with respect to the ground-truth humorous sentence. This feedback mechanism enables the student to refine its outputs without collapsing toward the stylistic distribution of the guiding model.

Formally, the loss function can be written as in Equation 1 when the LLM guide C_θ is frozen ($\mathcal{L}(\phi)$) and as in Equation 2 when the LLM guide is trainable ($\mathcal{L}(\phi, \theta)$). Each learning example from the dataset is comprised of a setup (s) and its continuation, termed the punchline (p). The setup, concatenated with its punchline, constitutes a humorous sentence (h).

$$\mathcal{L}(\phi) = \min_{\phi} \Big[\mathcal{L}_{\text{gen}}(G_\phi(h); h) + \mathcal{L}_{\text{cor}}(C_\theta(h_{\text{cor}}); h) \Big] \quad (1)$$

$$\begin{aligned} \mathcal{L}(\phi, \theta) = &\min_{\phi} \Big[\mathcal{L}_{\text{gen}}(G_\phi(h); h) \Big] \\ &+ \min_{\phi, \theta} \Big[\mathcal{L}_{\text{cor}}(C_\theta(h_{\text{cor}}); h) \Big] \\ &+ \min_{\theta} \Big[\mathcal{L}_{\text{guide}}(C_\theta(h); h) \Big] \end{aligned} \quad (2)$$

The generator loss ($\mathcal{L}_{gen}$) is the causal language modeling (CLM) loss, training the generator to reconstitute a humorous sentence ($\hat{h}$) given the ground truth as input (h).

The correction loss ($\mathcal{L}_{cor}$) is applied to h_{cor}, which is the concatenation of the setup (s) and the generated punchline ($\hat{p} = G_\phi(s)$), such that

[2]Note that jokes and punchlines are used interchangeably. ϕ and θ denote the respective model parameters.

$h_{cor} = [s; \hat{p}]$, where $[;]$ refers to the concatenation operation. $\mathcal{L}_{cor}$ is the CLM loss, such that the guiding model reconstitutes the ground truth (h) based on the input h_{cor}.

The guiding loss ($\mathcal{L}_{guide}$) is the CLM loss, training the model C_θ to reconstitute a humorous sentence ($\hat{h}$) given the ground truth as input (h).

$\mathcal{L}(\phi)$ and $\mathcal{L}(\phi, \theta)$ refer to the two final loss terms. In the case of $\mathcal{L}(\phi)$, the error loss returned by the guide C_θ is exclusively propagated to the generator G_ϕ to optimize itself, while the guide weights remain frozen. In the case of $\mathcal{L}(\phi, \theta)$, the error is propagated both in the guide and the generator.

Achieving differentiability in the optimization over generator parameters ϕ, crucial for the sampled $\hat{p} = G_\phi(s)$, is facilitated by the Gumbel-softmax (Jang et al., 2017).

5 Experimental Setup

Models and Baselines. We evaluate our proposed approach against several models and baselines, encompassing different model sizes and training strategies. All student models are based on GPT-2 (140M parameters) (Radford et al., 2019), and the large guide LLM is either Mistral-7B (7B parameters), LLaMA3-8B (8B parameters), or Qwen3-8B (8B parameters). Implemented baselines are given as follows.

Finetuned GPT-2 (FT-GPT2): The GPT-2 model finetuned directly on our curated setup-punchline joke dataset. This serves as our primary student model baseline.
Zero-Shot LLM (ZS-LLM): The LLM model used without any task-specific finetuning, generating punchlines in a zero-shot manner.
Finetuned LoRA LLM (LoRA-LLM): The LLM model finetuned on our curated setup-punchline joke dataset. This represents a strong baseline.
KD-Imitate ZS GPT-2 (LLM): The pre-trained GPT-2 model is trained to imitate punchlines generated by ZS-LLM for setups in our training data. This corresponds to the imitation phase of the framework in (Ravi et al., 2024).
KD-Imitate LoRA GPT-2 (LLM): Similar to the above, but GPT2 imitates punchlines generated by LoRA-LLM.
KD-Critique GPT-2 (LLM): The pre-trained GPT-2 model is first trained via imitation learning from ZS-LLM (as per KD-Imitate ZS (GPT-2 $\leftarrow$ ZS-LLM)) and is then further refined using Direct Preference Optimization (DPO) (Rafailov et al., 2023) with ZS-LLM as the preference provider. This methodology aligns with the two-stage distillation framework proposed in (Ravi et al., 2024), which is the best frugal model in the literature.
G-KD-Frozen GPT-2 (LLM): Our primary proposed framework where GPT2 is trained with guidance from a frozen, 4-bit quantized LLM model providing a corrective signal, as described by $L(\phi)$ in Section 4.
G-KD-LoRA GPT-2 (LLM): An ablation of our approach where the LLM guide model is also made parameter-efficiently trainable using LoRA alongside GPT2, as described by $L(\phi, \theta)$ in Section 4.

Evaluation Metrics. For evaluation, we employ a multi-faceted approach, with a great number of different metrics. In particular, we are the first to propose a diversity metric, which evaluates how much new models differ from the generic fine-tuned backbones (i.e. FT-GPT2 and LoRA-LLM).

Automatic Humor Classification: Following Weller and Seppi (2020), we use a BERT-based classifier (Devlin et al., 2019) trained on our full setup-punchline training dataset (not just the top 20,000 jokes) to classify generated setup+punchline pairs as humorous or not-humorous. We report Humor Accuracy.
N-gram Similarity Metrics: We compute BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) by comparing the generated punchline (appended to the ground truth setup) against the ground truth.
Semantic Similarity: We use BERTScore (Zhang et al., 2019) to measure the semantic similarity between the generated setup+punchline and the ground truth.
Diversity Metrics: To analyze the lexical diversity of generated punchlines relative to baseline models, we compute BLEU and ROUGE-L between the outputs of a target model and the outputs of a reference model (e.g., FT-GPT2 or LoRA-LLM) for the same set of setups. A lower score indicates greater lexical dissimilarity (higher diversity) from the reference model.
LLM-as-Judge Evaluation. To assess the quality of generated jokes beyond N-gram metrics, we employ a LLM-as-judge evaluation. We use GPT-4.1-mini[3] as the judge. For each pair of models to

[3] https://openai.com/index/gpt-4-1/

Model	Active Params (Train / Infer)	Curated Data (F)					Non-Curated Data (UF)				
		BLEU	ROUGE-L	BERTScore	Humor Acc.	Win Rate	BLEU	ROUGE-L	BERTScore	Humor Acc.	Win Rate
FT GPT-2	140M / 140M	0.3263	0.5248	0.8925	0.6675	0.4752	0.2633	0.4609	0.8631	0.6575	0.5264
KD-Critique GPT-2 (Qwen)	8.14B / 140M	0.3370	0.5130	0.9122	0.5732	0.4611	**0.3206**	**0.4982**	**0.9100**	0.6018	0.4897
KD-Imitate ZS GPT-2 (Mistral)	7.14B / 140M	**0.4186**	0.5660	**0.9228**	0.5462	0.4778	0.1910	0.3360	0.8844	0.5596	0.4752
KD-Imitate LoRA GPT-2 (Mistral)	7.14B / 140M	0.2508	0.4119	0.8726	**0.6683**	0.5127	0.1893	0.3364	0.8819	0.6353	0.1601
G-KD-Frozen GPT-2 (Mistral)	7.14B / 140M	0.3191	0.5204	0.8868	0.6678	**0.5938**	0.2459	0.4641	0.8616	0.6615	0.5689
G-KD-LoRA GPT-2 (Mistral)	7.14B / 140M	0.3538	**0.5679**	0.8952	0.6608	0.5629	0.2417	0.4811	0.8615	**0.6649**	0.5464

Table 1: Comparison of humor generation performance on curated (F) and non-curated (UF) data. Automatic metrics include BLEU, ROUGE-L, and BERTScore. Humor Accuracy and Win Rate are respectively obtained via a BERT-based classifier and LLM-as-Judge evaluation (44 models compared to compute win rate). Mean scores over 5 runs are reported. Only top-performing GPT-2-based configurations in terms of **humor accuracy** on the curated dataset are shown. Exhaustive results for all configurations are provided in Appendix with standard deviation values.

Model	Curated Data (F)					Non-Curated Data (UF)				
	BLEU	ROUGE-L	BERTScore	Humor Acc.	Win Rate	BLEU	ROUGE-L	BERTScore	Humor Acc.	Win Rate
LoRA Mistral	**0.3873**	**0.5744**	**0.9116**	0.4619	0.4115	0.2334	0.4002	0.8862	**0.6875**	**0.7459**
LoRA Qwen	0.2300	0.5389	0.8586	0.4848	0.5206	0.2295	**0.5492**	0.8512	0.5241	0.5246
LoRA LLaMA	0.2285	0.5015	0.8641	0.5928	**0.5786**	0.2399	0.5166	0.8608	0.5199	0.5489
G-KD-Frozen GPT-2 (Mistral)	0.3191	0.5204	0.8868	0.6678	0.5629	0.2459	0.4641	0.8616	0.6615	0.5464

Table 2: Comparison of G-KD-Frozen GPT-2 (Mistral) with LLM baselines on curated (F) and non-curated (UF) data. Automatic metrics include BLEU, ROUGE-L, and BERTScore. Humor Accuracy and Win Rate are obtained via a BERT-based classifier and an LLM-as-Judge evaluation (44 models compared to compute win rate). Mean scores over 5 runs are reported. Exhaustive results are provided in Appendix with standard deviation values.

be compared, we sample 500 unique setups from the test set. Both models generate a punchline for each setup. The LLM judge is then presented with the setup and the two generated punchlines (order randomized) and asked to choose which punchline is better in terms of humor and overall quality, following a multiple-choice prompting strategy as in (Ravi et al., 2024)[4]. The final reported metric is the Win Rate for a model, calculated as the percentage of times its generated punchline is preferred by the LLM judge over its competitor across all pairwise comparisons. We report the average Win Rate for each model against all other evaluated models.

6 Results

6.1 Joke Quality Evaluation

We first evaluate joke quality using automatic metrics and LLM-as-judge evaluations (cf. Table 1, Table 2 and Table 3). The non-curated (UF) data *only serves to assess generalization and models are not used to produce potentially harmful jokes*.

Automatic Quality Metrics. In Table 1, our approach, G-KD-Frozen GPT-2 (Mistral), achieves a humor accuracy of 66.78%, improving over FT-GPT2 (66.75%) and significantly outperforming KD baselines such as KD-Critique GPT-2 (Qwen) (57.32%) and KD-Imitate ZS GPT-2 (Mistral) (54.62%), demonstrating the effectiveness of our feedback-based mechanism over standard KD. It also competes with KD-Imitate LoRA GPT-2 (Mistral) (66.83%), the best performing model in terms of accuracy, but without statistical improvement compared to our approach. Moreover, our approach produces the best Win Rate score (59.38%) when models are a ranked by top humor accuracy, as well as N-gram (BLEU, ROUGE-L) and semantic (BERTScore) scores evidence clear superiority of G-KD-Frozen GPT-2 (Mistral) compared to KD-Imitate LoRA GPT-2 (Mistral).

In Table 2, the LoRA Mistral-7B shows the highest BLEU and ROUGE-L scores for all finetuned versions of LLMs. However, our G-KD-Frozen (Mistral) evidences drastically higher humor accuracy when compared to all LoRA LLM versions. This indicates that the LoRA Mistral-7B learns to copy jokes better, but its outputs are less humorous. Moreover, our approach competes in terms of Win Rate with all LoRA-based LLM versions, being LoRA LLaMA the only better model compared to our lightweight model.

LLM-as-Judge Evaluation. Results in Table 3 consistently demonstrate that our frozen LLM feed-

[4]The prompt is given in the Appendix.

Model	Active Params (Train / Infer)	Curated Data (F)					Non-Curated Data (UF)				
		BLEU	ROUGE-L	BERTScore	Humor Acc.	Win Rate	BLEU	ROUGE-L	BERTScore	Humor Acc.	Win Rate
FT GPT-2	140M / 140M	0.3263	0.5248	0.8925	0.6675	0.4752	0.2633	0.4609	0.8631	0.6575	0.5264
KD-Critique GPT-2 (Qwen)	8.14B / 140M	0.3370	0.5130	0.9122	0.5732	0.4611	0.3206	0.4982	0.9100	0.6018	0.4897
KD-Imitate ZS GPT-2 (LLaMA)	8.14B / 140M	0.3351	0.5196	**0.9129**	0.5661	0.4721	**0.3564**	**0.5409**	**0.9143**	0.5162	0.4521
KD-Imitate LoRA GPT-2 (Mistral)	7.14B / 140M	0.2508	0.4119	0.8726	**0.6683**	0.5127	0.1893	0.3364	0.8819	0.6353	0.1601
G-KD-Frozen GPT-2 (Mistral)	7.14B / 140M	0.3191	0.5204	0.8868	0.6678	**0.5938**	0.2459	0.4641	0.8616	0.6615	0.5689
G-KD-LoRA GPT-2 (LLaMA)	8.14B / 140M	**0.3386**	**0.5472**	0.8865	0.6421	**0.6063**	0.2466	0.4598	0.8637	**0.6988**	0.6372

Table 3: Comparison of humor generation performance on curated (F) and non-curated (UF) data. Automatic metrics include BLEU, ROUGE-L, and BERTScore. Humor Accuracy and Win Rate are respectively obtained via a BERT-based classifier and LLM-as-Judge evaluation (44 models compared to compute win rate). Mean scores over 5 runs are reported. Only top-performing GPT-2-based configurations in terms of **win rate** on the curated dataset are reported for all possible model alternatives. Exhaustive evaluation results for all configurations are provided in Appendix with standard deviation values.

back mechanism G-KD-Frozen GPT-2 (Mistral) enhances joke quality in the compact GPT-2 model series, surpassing both the FT GPT-2 baseline and most KD alternatives in humor accuracy and LLM-as-a-judge preferences. In particular, G-KD-Frozen GPT-2 (resp. G-KD-LoRA GPT-2) achieves a win rate of 59.38% (resp. 60.63%), when the best performing KD approach onlyy reaches 51.27%. This tends to suggest that guiding the student with corrective feedback is a more effective way of transferring knowledge than solely imitating teacher outputs or applying post-imitation DPO critique as in KD-Critique GPT-2 (Qwen). Our approach seems to efficiently utilize the capabilities of the LLM without direct mimicry, potentially mitigating issues like homogenization seen in some KD approaches (Lee et al., 2025).

6.2 Analysis of Output Diversity

Beyond joke quality, we analyze the diversity of the generated punchlines in Table 4 . We assess lexical and semantic dissimilarity of model outputs compared to two reference models: the baseline FT-GPT-2 and the best teacher LoRA Mistral[5],FT Gemma-1B, and FT Qwen-8B.For the additional Gemma-based experiments, we report diversity results on the curated dataset (F). Lower BLEU, ROUGE-L, and BERTScore values indicate greater dissimilarity (i.e., higher diversity) from the respective reference model's outputs.

Diversity from FT-GPT-2. All knowledge distillation methods[6] exhibit low N-gram similarity to FT-GPT-2. Their BLEU scores indicate a strong divergence from the student's original finetuned model, likely because these methods heavily adopt the stylistic characteristics of the teacher, which are inherently different from GPT-2. Their BERTScore values are also steadily lower than our methods, reflecting broader stylistic and semantic shift induced by the teacher during imitation. Comparatively, our approaches G-KD-Frozen GPT-2 and G-KD-LoRA GPT-2 are more conservative with respect to the original backbone.The additional Gemma-based configurations also show clear dissimilarity from FT-GPT-2. The standalone Gemma baselines exhibit lower similarity, while the guided Gemma variants remain relatively closer to FT-GPT-2 but still show substantial lexical and semantic divergence.

Diversity from LoRA Mistral. When comparing outputs to the teacher LoRA Mistral, all models (not directly trained to mimic LoRA Mistral), such as G-KD-Frozen GPT-2 (Mistral) and G-KD-LoRA GPT-2 (Mistral), show lower semantic similarity with respect to LoRA Mistral, as evidenced by their BERTScores (resp. 87.97% and 88.21%). This reflects higher dissimilarity and, therefore, greater diversity from the teacher model in terms of semantics. However, KD-Imitate LoRA GPT-2 (Mistral) tends to show the highest lexical diversity with lower BLEU and ROUGE-L score, clearly evidencing a high diversity when compared to both GPT-2 and Mistral backbones overall.When compared with FT Qwen-8B, both guided Gemma variants show low lexical and semantic similarity. This indicates that, despite receiving corrective guidance from Qwen, the Gemma generators retain substantial independence from the guide model.

[5]We only present results for Mistral as this is the best LLM for our approach.

[6]We present all best models as defined in Table 1 and Table 3.

Model	Curated (F) vs FT GPT-2			Non-Curated (UF) vs FT GPT-2			Curated (F) vs LoRA Mistral-7B			Non-Curated (UF) vs LoRA Mistral-7B			Curated (F) vs FT Gemma-1B			Curated (F) vs FT Qwen-8B		
	BLEU	ROUGE-L	BERTScore	BLEU	ROUGE-L	BERTScore	BLEU	ROUGE-L	BERTScore	BLEU	ROUGE-L	BERTScore	BLEU	ROUGE-L	BERTScore	BLEU	ROUGE-L	BERTScore
KD-Critique GPT-2 (Qwen)	0.3393	0.4463	**0.8758**	0.3393	0.4354	0.8738	–	–	–	–	–	–	–	–	–	–	–	–
KD-Critique GPT-2 (Mistral)	–	–	–	–	–	–	0.3623	0.4756	0.8853	**0.2104**	**0.3223**	0.8557	–	–	–	–	–	–
KD-Imitate ZS GPT-2 (Mistral)	0.3291	0.4797	0.8843	0.2121	0.3290	**0.8365**	0.3588	0.4881	0.8890	0.2144	0.3267	0.8609	–	–	–	–	–	–
KD-Imitate ZS GPT-2 (LLaMA)	0.3390	0.4476	0.8769	0.3355	0.4638	0.8776	–	–	–	–	–	–	–	–	–	–	–	–
KD-Imitate LoRA GPT-2 (Mistral)	**0.3140**	**0.4430**	0.8828	**0.2112**	**0.3259**	0.8380	**0.2865**	**0.4363**	0.8836	0.2204	0.3278	0.8583	–	–	–	–	–	–
G-KD-Frozen GPT-2 (Mistral)	0.4852	0.5995	0.9111	0.2736	0.4212	0.8703	0.3420	0.4713	**0.8797**	0.2487	0.3498	**0.8398**	–	–	–	–	–	–
G-KD-LoRA GPT-2 (Mistral)	0.4024	0.5361	0.8970	0.4024	0.5361	0.8970	0.3728	0.4936	0.8821	0.2468	0.3462	0.8379	–	–	–	–	–	–
G-KD-LoRA GPT-2 (LLaMA)	0.4449	0.5661	0.9005	0.2981	0.4576	0.8736	–	–	–	–	–	–	–	–	–	–	–	–
LoRA Fine-tuned Gemma-1B	0.2254	0.4181	0.8607	–	–	–	0.1972	0.3490	0.8660	–	–	–	–	–	–	–	–	–
Zero-shot Gemma-1B	0.2396	0.4035	0.8680	–	–	–	0.2095	0.4190	0.8338	–	–	–	–	–	–	–	–	–
G-KD-Frozen F Gemma (Qwen)	0.3254	0.5181	0.8845	–	–	–	0.3177	0.4490	0.8860	–	–	–	0.2173	0.3291	0.8572	0.2682	0.3901	0.8352
G-KD-LoRA F Gemma (Qwen)	0.3396	0.5035	0.8807	–	–	–	0.3392	0.4719	0.8638	–	–	–	0.2178	0.3220	0.8627	0.2866	0.3810	0.8437

Table 4: Evaluation of diversity based on automatic metrics (BLEU, ROUGE-L, and BERTScore) computed against generations from the corresponding fine-tuned teacher. Results are averaged over 5 runs. Lower similarity indicates greater diversity. Most diverse models (i.e., lowest similarity scores) are shown in **bold**.

6.3 Human Evaluation

We assigned 300 test samples including six different models to two annotators with a linguistic background. The annotators were unaware of the model that produced the generated text. They were instructed to rate the generated jokes on a Likert scale ranging from 1 to 5 based on two criteria: funniness[7] and fluency[8]. Table 5 presents overall results, showing that G-KD-Frozen GPT-2 (Mistral) outperforms FT-GPT-2, although still scoring lower than ZS-Mistral or LoRA Mistral, but with 98% fewer parameters. It is interesting to notice that both funniness and fluency are improved by the introduction of the guiding generation, with an increased performance of 22.39% and 17.11% respectively. Moreover, our approach beats KD-Critique GPT-2 (Mistral) in funniness, the most sophisticated model proposed by (Ravi et al., 2024). However, our experiments show that KD-Imitate ZS GPT-2 (Qwen) is a strong alternative for lightweight models based on human annotation, although automatic metrics and LLM-as-a-judge show poor performances for this particular model, strengthening the fact that the evaluation of humor generation is a difficult task. Inter-annotator agreement metrics (e.g. Cohen's κ (James, 2026) of 0.571 (resp. 0.596) for funniness (resp. fluency) - see Appendix A.1 for more detailed metrics (James, 2026; Bobicev and Sokolova, 2017)) evidence moderate-to-substantial agreement between annotators verifying that humor generation is a subjective task.

Models	Funniness	Fluency
G-KD-Frozen GPT-2 (Mistral)	3.28	3.56
KD-Critique GPT-2 (Mistral)	3.25	3.82
KD-Imitate ZS GPT-2 (Qwen)	3.33	**4.01**
ZS-Mistral	**3.56**	4.00
FT-GPT-2	2.68	3.04
LoRA-Mistral	3.38	3.71

Table 5: Results for human evaluation based on 300 test samples from the curated dataset (F) processed by 2 annotators. Best results are in bold while second best ones are underlined.

6.4 Error Analysis

While positive results can be obtained (see four examples in the Appendix), the generation process also presents different flaws, which are addressed as follows. Examples are given in Table 6.

i) *Background Incongruity:* (1st Example) The model often fails to create punchlines relevant to the setup background context, leading to non-humorous outputs.

ii) *Polysemy Misinterpretation:* (2nd example) The model struggles with *polysemy*. Here, it misses the double meaning of *inflation* (economic vs. tire inflation), resulting in less humorous punchlines.

iii) *OOD Generalization:* (3rd example) Lack of out-of-domain (OOD) generalization is observed, as the model fails to recognize subtle connotations in setups. The model seems to be unaware of the term "globetrotter" and its general meaning.

iv) *Absent Wordplay:* (4th example) The model fails to capture the intended wordplay. The humor relies on the dual meaning of "web" (spider web vs. web development), but the generated punchline remains literal resulting in a non-funny quote.

6.5 Discussion of Research Questions

In this section, we revisit the research questions posed in the introduction and discuss how our findings address them.

R1: Can a compact student model, guided by a frozen LLM, strike a balance between humor quality and diversity?

Our results confirm that a guided compact model

[7]Scores range from 1 for unfunny punchlines to 5 for the best ones.

[8]Scores range from 1 for inarticulate answers to 5 for natural and regular responses.

Setup	Punchline	G-KD-Frozen GPT-2 (Mistral)	Reason
C.S. Lewis once wrote an anthology on anime culture	The Chronicles of Nani-a.	It was a work of pure talent.	Background Incongruity
Stopped to put air in my tires today. The pump cost $1.50! I remember when those things used to only cost 25 cents.	Guess the price has adjusted for inflation.	I'm so glad I was in the first place.	Polysemy
What do you call a globetrotter after you put him in a blender?	A Harlem Shake.	A cashew.	OOD Generalization
What does Bob the spider do for a living?	He's a web developer.	He makes webs.	Absent Wordplay

Table 6: Examples of error cases made by G-KD-Frozen GPT-2 (Mistral).

can achieve this balance. The proposed G-KD-Frozen GPT-2 (Mistral) method significantly enhances humor quality over the FT-GPT2 baseline and KD methods. This is evidenced by superior automatic humor accuracy (66.78% vs. 66.75% for FT-GPT-2), higher LLM-as-Judge win rates (59.38% vs. 47.52% for FT-GPT-2), and improved human-rated funniness (3.28 vs. 2.68 for FT-GPT-2). Concurrently, our diversity analysis (see Table 4) indicates that G-KD-Frozen GPT-2 (Mistral) generates outputs distinct from FT-GPT-2, fostering an expansion of generative capabilities. While KD methods show greater lexical divergence from FT-GPT-2, our approach achieves this diversity without the marked drop in humor quality characteristic of KD techniques, thus striking a more effective balance.

R2: Can our approach offer better efficiency and controllability while maintaining comparable or superior humor quality to standard fine-tuning and KD methods?

In terms of efficiency, G-KD-Frozen GPT-2 (Mistral) trains the compact 140M GPT-2 student while leveraging a frozen, 4-bit quantized Mistral-7B guide. This ensures that inference is performed solely by the lightweight student, maintaining low computational costs comparable to the Mistral-7B baselines and KD-trained students. The efficiency gain over needing to fine-tune or constantly query a large LLM is significant based on the automatic evaluation metric and the LLM-as-a-judge. However, the manual evaluation evidences more mitigated results, clearly evidencing the difficulty in evaluating humor generation.

7 Conclusions

In this paper, we introduce a feedback-based framework for humor generation in which a compact generator model is guided by a LLM providing corrective feedback. The guiding model can be instantiated using different LLM architectures such as Mistral, LLaMA, and Qwen under either frozen or learnable configurations. Among these variants, the configuration using a frozen Mistral-7B guide (G-KD-Frozen GPT-2 (Mistral)) achieves strong performances in our experiments. Unlike standard distillation, our method provides corrective supervision without requiring behavioral mimicry. Through comprehensive evaluation, including automatic metrics, human annotation, diversity analysis, and LLM-as-a-judge scoring, we show that our approach outperforms a wide range of KD baselines in quality and diversity, while remaining highly efficient in terms of humor production.

Limitations

While our proposed framework demonstrates promising results, there are certain limitations. First, our study primarily focuses on setup-punchline jokes sourced from Reddit. Consequently, the model's performance and the nature of the generated humor may not directly generalize to other humor styles (e.g., satire, storytelling-based humor) or cultural contexts not well-represented in this dataset. Second, the effectiveness of our guidance mechanism is inherently tied to the capabilities of the frozen LLM (Mistral-7B). Although Mistral-7B is a strong guide, it is not a perfect oracle. Its own understanding or generation biases for humor could inadvertently influence the student model, potentially limiting the exploration of certain humorous avenues or, as shown in our error analysis (Section 6.4), occasionally providing suboptimal corrective signals for highly nuanced jokes. Finally, the evaluation of true novelty and creativity in humor remains a challenging open problem, and our current metrics provide a partial view.

Ethical Considerations

Humor generation presents particular ethical challenges because humorous content can easily involve stereotypes, offensive language, bullying, hateful conduct, profanity, or culturally sensitive material. To reduce these risks, we construct a curated version of our Reddit-based setup–punchline dataset in which potentially harmful jokes are auto-

matically filtered using an LLM-based safety classifier. Filtering is applied to both the training and test sets. The non-curated split is retained only to evaluate the generalization capability of the proposed method and is neither intended for deployment nor released, in order to reduce the risk of enabling harmful content generation. Despite this filtering process, automated safety classification is not perfect and may fail to identify subtle, context-dependent, or culturally specific forms of harmful humor. Conversely, some benign jokes may also be incorrectly removed. Therefore, the curated dataset should not be interpreted as completely free from harmful or biased content. In addition, the behavior of the student model depends partly on the guiding LLM, whose own biases, limitations, and cultural assumptions may influence the corrective feedback provided during training. The generated outputs are evaluated using automatic metrics, LLM-based judgments, and human annotation. Because humor is subjective and culturally dependent, these evaluations provide only a partial view of humor quality and appropriateness. The proposed framework is intended for research on lightweight humor generation and should be used with appropriate safeguards in any real-world deployment, particularly in applications involving diverse users or sensitive contexts.

Acknowledgements

This work was carried out using the computing resources provided by CRIANN. Dibyanayan Bandyopadhyay gratefully acknowledges support from the Prime Minister's Research Fellowship (PMRF) under Grant No. 2703564. The authors also thank the anonymous reviewers for their constructive feedback and valuable suggestions.

References

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12:681–699.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263.

Miriam Amin and Manuel Burghardt. 2020. 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, Online. International Committee on Computational Linguistics.

Aseem Arora, Gaël Dias, Adam Jatowt, and Asif Ekbal. 2022. Transfer learning for humor detection by twin masked yellow Muppets. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1–7, Online only. Association for Computational Linguistics.

Oswald Barral, Ilkka Kosunen, and Giulio Jacucci. 2017. No need to laugh out loud: Predicting humor appraisal of comic strips based on physiological signals in a realistic environment. *ACM Trans. Comput.-Hum. Interact.*, 24(6).

Kim Binsted and Graeme Ritchie. 1994. An implemented model of punning riddles. *Preprint*, arXiv:cmp-lg/9406022.

Victoria Bobicev and Marina Sokolova. 2017. Inter-annotator agreement in sentiment analysis: Machine learning perspective. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 97–102, Varna, Bulgaria. INCOMA Ltd.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024a. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*. Featured Certification.

Yuyan Chen, Yichen Yuan, Panjun Liu, Dayiheng Liu, Qinghao Guan, Mengfei Guo, Haiming Peng, Bang Liu, Zhixu Li, and Yanghua Xiao. 2024b. Talk funny! a large-scale humor response dataset with chain-of-humor interpretation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pages 32694–32717.

Joseph H. F. James. 2026. Counting on consensus: Selecting the right inter-annotator agreement metric for NLP annotation and evaluation. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 4434–4446, Palma de Mallorca, Spain. ELRA Language Resource Association.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Sophie Jentzsch and Kristian Kersting. 2023. ChatGPT is fun, but it is not funny! humor is still challenging large language models. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 325–340, Toronto, Canada. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 24872–24895. PMLR.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. RankGen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 199–232, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sunbowen Lee, Junting Zhou, Chang Ao, Kaige Li, Xeron Du, Sirui He, Haihong Wu, Tianci Liu, Jiaheng Liu, Hamid Alinejad-Rokny, Min Yang, Yitao Liang, Zhoufutu Wen, and Shiwen Ni. 2025. Quantification of large language model distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4985–5004, Vienna, Austria. Association for Computational Linguistics.

Jens Lemmens and Victor De Marez. 2026. Computational humor modeling: A survey on the state of the art. *ACM Comput. Surv.*, 58(7).

Jianquan Li, XiangBo Wu, Xiaokang Liu, Qianqian Xie, Prayag Tiwari, and Benyou Wang. 2023. Can language models make fun? a case study in Chinese comical crosstalk. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7581–7596, Toronto, Canada. Association for Computational Linguistics.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6121–6133, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tyler Loakman, William Thorne, and Chenghua Lin. 2025. Who's laughing now? an overview of computational humour generation and explanation. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 780–794, Hanoi, Vietnam. Association for Computational Linguistics.

Junyu Lu, Kai Ma, Kaichun Wang, Kelaiti Xiao, Roy Ka-Wei Lee, Bo Xu, Liang Yang, and Hongfei Lin. 2025. Is LLM an overconfident judge? unveiling the capabilities of LLMs in detecting offensive language with annotation disagreement. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5609–5626, Vienna, Austria. Association for Computational Linguistics.

Piotr Mirowski, Juliette Love, Kory Mathewson, and Shakir Mohamed. 2024. A robot walks into a bar: Can language models serve as creativity supporttools for comedy? an evaluation of llms' humour alignment with comedians. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 1622–1636, New York, NY, USA. Association for Computing Machinery.

Ria Mundhra, Ting Jen Lim, Hong Ngan Duong, Kheng Hui Yeo, and Andreea I. Niculescu. 2021. *Towards a Humorous Chat-Bot Companion for Senior Citizens*, pages 31–39. Springer Singapore, Singapore.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Taivo Pungas. 2017. A dataset of english plaintext jokes.

Kexin Quan, Pavithra Ramakrishnan, and Jessie Chin. 2025. Can ai take a joke—or make one? a study

of humor generation and recognition in llms. In *Proceedings of the 2025 Conference on Creativity and Cognition*, C&C '25, page 431–437, New York, NY, USA. Association for Computing Machinery.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sahithya Ravi, Patrick Huber, Akshat Shrivastava, Vered Shwartz, and Arash Einolghozati. 2024. Small but funny: A feedback-driven approach to humor distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13078–13090, Bangkok, Thailand. Association for Computational Linguistics.

Alexey Tikhonov and Pavel Shtykovskiy. 2024. Humor mechanics: Advancing humor generation with multistep reasoning. In *Proceedings of the 15th International Conference on Computational Creativity, ICCC 2024, Jönköping, Sweden, June 17-21, 2024*, pages 31–41. Association for Computational Creativity (ACC).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Dmitry Vikhorev, Daria Galimzianova, Svetlana Gorovaia, Elizaveta Zhemchuzhina, and Ivan P. Yamshchikov. 2024. Cleancomedy: Creating friendly humor through generative techniques. *Preprint*, arXiv:2412.09203.

Han Wang, Yilin Zhao, Dian Li, Xiaohan Wang, sinbadliu, Xuguang Lan, and Hui Wang. 2025. Innovative thinking, infinite humor: Humor research of large language models through structured thought leaps. In *International Conference on Learning Representations*, volume 2025, pages 42061–42085.

Orion Weller and Kevin Seppi. 2020. The rJokes dataset: a large scale humor collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6136–6141, Marseille, France. European Language Resources Association.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660, Melbourne, Australia. Association for Computational Linguistics.

Alessandro Zangari, Matteo Marcuzzo, Andrea Albarelli, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2025. Pun unintended: LLMs and the illusion of humor understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27936–27971, Suzhou, China. Association for Computational Linguistics.

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

Shanshan Zhong, Zhongzhan Huang, Shanghua Gao, Wushao Wen, Liang Lin, Marinka Zitnik, and Pan Zhou. 2024. 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 (CVPR)*, pages 13246–13257.

Yftah Ziser, Elad Kravi, and David Carmel. 2020. Humor detection in product question answering systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 519–528, New York, NY, USA. Association for Computing Machinery.

A Appendix

Overall Results. Table 9 summarizes the automatic evaluation results for all model configurations on curated (F) and Non-curated (UF) data, together with the number of active parameters during training and inference. The FT-GPT-2 baseline represents the standard finetuned student, while the LoRA and ZS rows correspond to larger LLMs or zero-shot generators. KD-Critique and KD-Imitate variants capture imitation-based distillation baselines, and the G-KD-Frozen / G-KD-LoRA rows correspond to our feedback-guided methods.

The table illustrates that feedback-guided compact models (G-KD-Frozen GPT-2 and G-KD-LoRA GPT-2) achieve humor accuracy competitive

with or superior to KD-based methods while maintaining the same 140M active parameters at inference, in contrast to LoRA or ZS configurations that require billions of parameters. This supports our claim that corrective feedback allows small models to approach large-model humor quality without incurring large-model inference cost.

Diversity Evaluation. Table 10 reports BLEU, ROUGE-L, and BERTScore computed between each model's outputs and those of the FT-GPT-2 baseline. These scores quantify how closely different systems align with the finetuned GPT-2 punchline style. Higher similarity indicates that a method preserves the original GPT-2 behavior, whereas lower scores reflect greater stylistic and lexical divergence from FT-GPT-2.

Imitation-based KD methods tend to diverge more strongly from FT GPT-2, reflecting their adaptation toward the guide model's style, while our G-KD-Frozen and G-KD-LoRA models show moderate divergence: they move away from the baseline enough to enrich generative behavior, but without fully collapsing into guide imitation. This pattern is consistent with our goal of controlled stylistic expansion rather than complete stylistic transfer.

Table 11 presents BLEU, ROUGE-L, and BERTScore computed against LoRA Mistral-7B, which serves as the strong guide model in our framework. Here, higher scores indicate closer alignment to the guide model's punchline style, while lower scores suggest greater independence from the guide model.

As expected, imitation-based KD models exhibit relatively high similarity to LoRA Mistral, reflecting their objective of closely reproducing guide model outputs. In contrast, G-KD-Frozen and G-KD-LoRA maintain only moderate similarity: they benefit from guide feedback but do not fully mirror the guide model's behavior. Together with Table 10, this supports our analysis in Section 6.2 that feedback-guided generators occupy an intermediate regime between strict imitation and purely generator-driven generation.

LLM-as-a-judge. Table 12 reports the LLM-as-judge win rates for all models, where a higher score indicates that a model's punchlines are preferred more often over the FT GPT-2 baseline in pairwise comparisons. We distinguish between larger LLM generators (LoRA and ZS rows) and compact GPT-2–based generators (FT, KD, and G-KD variants).

The results show that G-KD-Frozen GPT-2 (Mistral) and G-KD-LoRA variants consistently achieve higher win rates than standard KD baselines, and in several cases rival or surpass LoRA-based LLMs. This aligns with the main paper's findings that corrective feedback from a frozen LLM provides a stronger training signal for humor generation than imitation-based distillation, enabling compact models to deliver humor quality competitive with much larger systems.

LLM-as-Judge Evaluation Protocol. To assess perceived humor quality beyond lexical overlap metrics, we adopt an LLM-as-judge evaluation framework. For each setup, two punchlines generated by competing models are presented to a strong external LLM acting as a judge. The punchlines are shown in randomized order to prevent positional bias, and model identities are hidden. The exact evaluation prompt is provided below.

The judge is prompted to select the punchline that is funnier while maintaining coherence, relevance to the setup, and natural language fluency. The evaluation prompt explicitly instructs the judge to prioritize humor quality over superficial similarity or length. The judge outputs a binary preference decision.

For each model comparison (Table 12), we evaluate all setups in the test set and compute the win rate as the proportion of times a model's punchline is preferred. A win rate greater than 0.50 indicates consistent preference over the baseline. To reduce stochastic effects, evaluations are conducted with fixed decoding parameters (temperature = 0) and repeated across multiple runs when necessary.

This protocol complements automatic metrics by capturing perceived humor effectiveness and creative impact, which are difficult to measure using n-gram overlap alone.

We provide the exact prompt used for LLM-as-a-judge evaluation to ensure reproducibility.

Qualitative Study. In Table 13, we show four different setups and their ground truth punchlines and compare them with generated punchlines from various models. Our approach produces more creative and funnier punchlines than those from baseline GPT-2 outputs while rivaling Mistral's humor quality. The frozen Mistral-7B guide helps the generator avoid common patterns and formulaic responses while maintaining coherence with the setup. The configuration that uses the Mistral-7B learnable guide, G-KD-LoRA GPT-2 (Mistral), also shows

LLM-as-a-Judge Prompt

You will be given two jokes, each consisting of a setup and a punchline. Choose the joke that is better overall as humor.

Evaluation Criteria:

- Funniness: Which joke is more humorous?
- Coherence: Which punchline better fits its setup?
- Naturalness: Which joke sounds more fluent and human-like?

Prioritize **funniness** most. Use coherence and naturalness as secondary criteria.

Option 1:
{setup + punchline}

Option 2:
{setup + punchline}

Output Format:
Reason: <*one concise sentence*>
Answer: <*1 or 2*>

Figure 3: Prompt used for pairwise evaluation in the LLM-as-a-Judge setting.

interesting results that can be on par with the frozen setting. For example, it happens that both models output the same joke. As expected, the guidance with a unique backbone model, here GPT-2, reveals less diversity infusion.

A.1 Inter-Annotator Agreement

To validate the reliability of our human evaluation (Section 6.3), we report inter-annotator agreement (IAA) between our two annotators across both evaluation dimensions: **Funniness** and **Fluency**. Following recommendations in the NLP annotation literature, we employ multiple complementary agreement metrics rather than relying on a single coefficient (James, 2026; Bobicev and Sokolova, 2017).

Metrics. We report the following five IAA measures.

(i) Raw Agreement is the simplest baseline, computed as the proportion of items on which the two annotators assign identical ratings:

$$A_o = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[r_{1,i} = r_{2,i}] \tag{3}$$

where N is the total number of annotated items and $r_{a,i}$ is the rating assigned by annotator a to item i. Raw agreement ignores chance-level coincidence and must therefore be complemented by chance-corrected metrics (James, 2026).

(ii) Cohen's κ (Bobicev and Sokolova, 2017; James, 2026) corrects for chance agreement under the assumption that both annotators draw independently from the same marginal category distribution:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{4}$$

where P_o is the observed agreement (Eq. 3) and P_e is the expected chance agreement:

$$P_e = \sum_k p_{1k} \cdot p_{2k} \tag{5}$$

with p_{ak} denoting the proportion of items annotator a assigned to category k. Bobicev and Sokolova (2017) show that on sentiment annotation tasks with ordinal scales, Cohen's κ provides a more reliable picture of genuine agreement than raw percentage alone.

(iii) $P_{\alpha\beta}$ and $P_{\alpha\bar{\beta}}$ (Agreement and Polar-Disagreement Proportions). Beyond a single scalar, it is informative to track both where annotators *agree exactly* ($P_{\alpha\beta}$) and where they *maximally disagree* ($P_{\alpha\bar{\beta}}$) (James, 2026). These are defined as:

$$P_{\alpha\beta} = \frac{\#\{i : r_{1,i} = r_{2,i}\}}{N} \tag{6}$$

$$P_{\alpha\bar{\beta}} = \frac{\#\{i : |r_{1,i} - r_{2,i}| = K - 1\}}{N} \tag{7}$$

where $K = 5$ is the number of Likert-scale points. A high $P_{\alpha\beta}$ paired with a low $P_{\alpha\bar{\beta}}$ indicates strong, consistent agreement; a non-negligible $P_{\alpha\bar{\beta}}$ signals systematic polarisation between annotators on at least a subset of items (James, 2026).

(iv) Weighted κ (Bobicev and Sokolova, 2017; James, 2026) extends Cohen's κ by penalising disagreements in proportion to their distance on the ordinal scale:

$$\kappa_w = 1 - \frac{\sum_{j,k} w_{jk}\, o_{jk}}{\sum_{j,k} w_{jk}\, e_{jk}} \tag{8}$$

where o_{jk} and e_{jk} are the observed and expected proportions of rating pairs in cell (j, k), and w_{jk} is a distance-based weight. We use quadratic weights $w_{jk} = 1 - \frac{(j-k)^2}{(K-1)^2}$, which give partial credit to near-misses and are the standard choice for ordinal Likert responses (Bobicev and Sokolova, 2017; James, 2026).

(v) Per-Label κ ($\kappa_{L\ell}$). James (2026) recommend computing agreement separately for each category label ℓ to expose systematic label-level biases that a global κ can mask. Each $\kappa_{L\ell}$ is a binary Cohen's κ where label ℓ is treated as the positive class and all remaining labels are merged into the negative class:

$$\kappa_{L\ell} = \frac{P_o^{(\ell)} - P_e^{(\ell)}}{1 - P_e^{(\ell)}} \tag{9}$$

The macro-average $\kappa_{\text{Avg}} = \frac{1}{K}\sum_{\ell=1}^{K} \kappa_{L\ell}$ provides a single summary of per-label agreement across the full scale.

Results. Table 7 reports the complete IAA profile for both annotation dimensions.

Interpretation. Both dimensions yield Cohen's κ in the range $[0.55, 0.60]$, which James (2026) classify as *moderate-to-substantial* agreement. The weighted κ values exceed 0.77 for both dimensions, indicating that even where exact label agreement is absent, the two annotators tend to be close on the ordinal scale a desirable property for Likert-based evaluation (Bobicev and Sokolova, 2017).

The $P_{\alpha\bar{\beta}}$ scores are low for both Funniness (0.36) and Fluency (0.29), confirming that polar disagreements cases where one annotator assigns the minimum and the other the maximum rating are uncommon. Following James (2026), this combination of moderate κ with a low polar-disagreement rate provides stronger evidence of annotation reliability than a high raw agreement figure alone, since it rules out the scenario in which annotators superficially agree on neutral labels while diverging sharply on the most salient items.

The per-label breakdown reveals that agreement is highest at the extremes of the scale ($\kappa_{L5} = 0.799$ for Funniness; $\kappa_{L2} = 0.673$ for Fluency), while mid-scale labels particularly κ_{L3}—are harder to agree on. Bobicev and Sokolova (2017) observe the same pattern in sentiment annotation: boundary labels are inherently more ambiguous, and annotators diverge most around the centre of an ordinal scale. Overall, the IAA results support the validity of our human evaluation conclusions reported in Section 6.3.

A.2 Correlation Between Automatic and Human Evaluation Metrics

Following prior work such as (Adlakha et al., 2024), which studies the correlation between proxy metrics and human judgments, we conduct a model-level correlation analysis using the six models included in our human evaluation.

Specifically, we compute Spearman's ρ and Kendall's τ between each automatic metric (BLEU, ROUGE-L, BERTScore, Humor Accuracy, and LLM-as-Judge Win Rate) and the human ratings for funniness and fluency.

The correlation results in Table 8 show that different evaluation metrics capture different aspects of humor generation quality. BLEU and ROUGE-L mainly reflect lexical overlap with the reference punchline, while BERTScore measures semantic similarity. However, in humor generation, a punchline can be funny even when it differs substantially from the reference. This helps explain why BLEU and ROUGE-L show weak or negative correlation with human funniness: higher reference overlap does not necessarily imply a funnier or more creative punchline.

Similarly, Humor Accuracy provides a useful supporting signal, but it does not fully capture subjective qualities such as funniness, creativity, fluency, and setup relevance. Its correlation with human funniness is positive but weak, and its relation with fluency is inconsistent. This further suggests that Humor Accuracy should not be used as the main evidence of humor generation quality for the proposed method, but rather as a supporting metric indicating whether the generated outputs exhibit humorous characteristics.

Overall, these results indicate that the different evaluation metrics provide complementary signals for assessing humor generation quality. In particular, human funniness ratings and LLM-as-Judge Win Rate show more substantial improvements over FT-GPT2 than Humor Accuracy, highlighting aspects of generation quality that are not fully captured by the automatic humor classifier.

#	Dimension	Pair	N	Raw (%)	κ	$P_{\alpha\beta}$	$P_{\alpha\bar{\beta}}$	κ_w	κ_{L1}	κ_{L2}	κ_{L3}	κ_{L4}	κ_{L5}	κ_{Avg}
1	**Funniness**	A1 vs A2	300	72.46	0.571	0.7246	0.3582	0.7776	0.6299	0.5376	0.5448	0.6876	0.7990	0.6398
2	**Fluency**	A1 vs A2	300	71.32	0.596	0.7132	0.2901	0.8423	0.6598	0.6731	0.5435	0.5561	0.6923	0.6250

Table 7: Inter-annotator agreement for human evaluation on the curated dataset (F). Raw agreement is reported as a percentage. κ: Cohen's κ (James, 2026); $P_{\alpha\beta}$: proportion of exact agreements (Eq. 6); $P_{\alpha\bar{\beta}}$: proportion of polar disagreements with Likert distance equals to $K-1=4$ (Eq. 7); κ_w: weighted κ with quadratic weights (Bobicev and Sokolova, 2017); $\kappa_{L\ell}$: per-label κ for Likert levels 1–5 (Eq. 9); κ_{Avg}: macro-average of per-label κ. $N = 3323$ annotation pairs per dimension.

Proxy Metric	Human Funniness Spearman's ρ	Human Funniness Kendall's τ	Human Fluency Spearman's ρ	Human Fluency Kendall's τ
BLEU	-0.143	-0.067	0.029	-0.067
ROUGE-L	-0.371	-0.333	-0.486	-0.333
BERTScore	-0.371	-0.333	0.257	0.200
Humor Accuracy	0.143	0.067	-0.029	0.067
LLM-as-Judge Win Rate	0.086	0.067	0.029	0.067

Table 8: Model-level correlation between automatic and LLM-based proxy metrics and human funniness/fluency ratings for the six models included in the human evaluation.

Model	Active Params (Train / Infer)	BLEU	ROUGE-L	BERTScore	Humor Accuracy
FT F GPT-2	140M / 140M	0.3263 ± 0.0005	0.5248 ± 0.0010	0.8925 ± 0.0004	0.6675 ± 0.0031
LoRA F Mistral	7.14B / 140M	0.3873 ± 0.0010	0.5744 ± 0.0008	0.9116 ± 0.0003	0.4619 ± 0.0025
LoRA F Qwen	8.14B / 140M	0.2300 ± 0.0006	0.5389 ± 0.0005	0.8586 ± 0.0003	0.4848 ± 0.0063
LoRA F LLaMA	8.14B / 140M	0.2285 ± 0.0006	0.5015 ± 0.0007	0.8641 ± 0.0002	0.5928 ± 0.0028
FT UF GPT-2	140M / 140M	0.2633 ± 0.0002	0.4609 ± 0.0003	0.8631 ± 0.0001	0.6575 ± 0.0016
LoRA UF Mistral	7.14B / 140M	0.2334 ± 0.0002	0.4002 ± 0.0001	0.8862 ± 0.0000	0.6875 ± 0.0010
LoRA UF Qwen	8.14B / 140M	0.2295 ± 0.0009	0.5492 ± 0.0009	0.8512 ± 0.0002	0.5241 ± 0.0052
LoRA UF LLaMA	8.14B / 140M	0.2399 ± 0.0000	0.5166 ± 0.0001	0.8608 ± 0.0000	0.5199 ± 0.0012
ZS Mistral	0 / 7B	0.2103 ± 0.0001	0.3630 ± 0.0002	0.8848 ± 0.0000	0.7112 ± 0.0002
ZS Qwen	0 / 8B	0.1905 ± 0.0003	0.3438 ± 0.0007	0.8858 ± 0.0001	0.6779 ± 0.0045
ZS LLaMA	0 / 8B	0.3131 ± 0.0018	0.5027 ± 0.0019	0.9089 ± 0.0003	0.4687 ± 0.0079
KD-Critique F GPT-2 (Mistral) (Ravi et al., 2024)	7.14B / 140M	0.3849 ± 0.0010	0.5450 ± 0.0006	0.9184 ± 0.0001	0.5111 ± 0.0039
KD-Critique F GPT-2 (Qwen)	8.14B / 140M	0.3370 ± 0.0011	0.5130 ± 0.0005	0.9122 ± 0.0001	0.5732 ± 0.0026
KD-Critique F GPT-2 (LLaMA)	8.14B / 140M	0.3146 ± 0.0015	0.5123 ± 0.0006	0.9082 ± 0.0002	0.5260 ± 0.0048
KD-Critique UF GPT-2 (Mistral) (Ravi et al., 2024)	7.14B / 140M	0.1908 ± 0.0001	0.3375 ± 0.0002	0.8842 ± 0.0001	0.5559 ± 0.0010
KD-Critique UF GPT-2 (Qwen)	8.14B / 140M	0.3206 ± 0.0008	0.4982 ± 0.0009	0.9100 ± 0.0001	0.6018 ± 0.0058
KD-Critique UF GPT-2 (LLaMA)	8.14B / 140M	0.3271 ± 0.0013	0.5172 ± 0.0009	0.9100 ± 0.0002	0.5177 ± 0.0048
KD-Imitate ZS F GPT-2 (Mistral) (Ravi et al., 2024)	7.14B / 140M	0.4186 ± 0.0007	0.5660 ± 0.0003	0.9228 ± 0.0001	0.5462 ± 0.0050
KD-Imitate ZS F GPT-2 (Qwen)	8.14B / 140M	0.3351 ± 0.0014	0.5196 ± 0.0009	0.9129 ± 0.0002	0.5661 ± 0.0032
KD-Imitate ZS F GPT-2 (LLaMA)	8.14B / 140M	0.3351 ± 0.0014	0.5196 ± 0.0009	0.9129 ± 0.0002	0.5661 ± 0.0032
KD-Imitate ZS UF GPT-2 (Mistral) (Ravi et al., 2024)	7.14B / 140M	0.1910 ± 0.0000	0.3360 ± 0.0002	0.8844 ± 0.0000	0.5596 ± 0.0023
KD-Imitate ZS UF GPT-2 (Qwen)	8.14B / 140M	0.3431 ± 0.0008	0.5226 ± 0.0008	0.9137 ± 0.0002	0.5833 ± 0.0031
KD-Imitate ZS UF GPT-2 (LLaMA)	8.14B / 140M	0.3564 ± 0.0028	0.5409 ± 0.0011	0.9143 ± 0.0004	0.5162 ± 0.0052
KD-Imitate LoRA F GPT-2 (Mistral) (Ravi et al., 2024)	7.14B / 140M	0.2508 ± 0.0002	0.4119 ± 0.0004	0.8726 ± 0.0001	0.6683 ± 0.0058
KD-Imitate LoRA F GPT-2 (Qwen)	8.14B / 140M	0.2193 ± 0.0006	0.4772 ± 0.0007	0.8559 ± 0.0003	0.4404 ± 0.0036
KD-Imitate LoRA F GPT-2 (LLaMA)	8.14B / 140M	0.2144 ± 0.0003	0.5013 ± 0.0008	0.8483 ± 0.0002	0.4441 ± 0.0016
KD-Imitate LoRA UF GPT-2 (Mistral) (Ravi et al., 2024)	7.14B / 140M	0.1893 ± 0.0001	0.3364 ± 0.0000	0.8819 ± 0.0000	0.6353 ± 0.0002
KD-Imitate LoRA UF GPT-2 (Qwen)	8.14B / 140M	0.2063 ± 0.0003	0.4902 ± 0.0015	0.8557 ± 0.0002	0.5125 ± 0.0041
KD-Imitate LoRA UF GPT-2 (LLaMA)	8.14B / 140M	0.2081 ± 0.0004	0.4769 ± 0.0013	0.8586 ± 0.0001	0.5685 ± 0.0055
G-KD-Frozen F GPT-2 (Mistral)	7.14B / 140M	0.3191 ± 0.0005	0.5204 ± 0.0012	0.8868 ± 0.0005	0.6678 ± 0.0061
G-KD-Frozen F GPT-2 (Qwen)	8.14B / 140M	0.3278 ± 0.0004	0.5258 ± 0.0004	0.8919 ± 0.0003	0.6555 ± 0.0029
G-KD-Frozen F GPT-2 (LLaMA)	8.14B / 140M	0.3246 ± 0.0014	0.5259 ± 0.0013	0.8931 ± 0.0005	0.6454 ± 0.0043
G-KD-Frozen UF GPT-2 (Mistral)	7.14B / 140M	0.2459 ± 0.0002	0.4641 ± 0.0002	0.8616 ± 0.0000	0.6615 ± 0.0037
G-KD-Frozen UF GPT-2 (Qwen)	8.14B / 140M	0.2612 ± 0.0008	0.4588 ± 0.0004	0.8656 ± 0.0002	0.6798 ± 0.0044
G-KD-Frozen UF GPT-2 (LLaMA)	8.14B / 140M	0.2466 ± 0.0004	0.4597 ± 0.0006	0.8637 ± 0.0001	0.6988 ± 0.0018
G-KD-LoRA F GPT-2 (Mistral)	7.14B / 140M	0.3538 ± 0.0018	0.5679 ± 0.0012	0.8952 ± 0.0005	0.6608 ± 0.0008
G-KD-LoRA F GPT-2 (Qwen)	8.14B / 140M	0.4075 ± 0.0031	0.6057 ± 0.0008	0.9131 ± 0.0003	0.6521 ± 0.0032
G-KD-LoRA F GPT-2 (LLaMA)	8.14B / 140M	0.3386 ± 0.0010	0.5472 ± 0.0012	0.8865 ± 0.0003	0.6421 ± 0.0032
G-KD-LoRA UF GPT-2 (Mistral)	7.14B / 140M	0.2417 ± 0.0005	0.4811 ± 0.0005	0.8615 ± 0.0000	0.6608 ± 0.0008
G-KD-LoRA UF GPT-2 (Qwen)	8.14B / 140M	0.2595 ± 0.0013	0.4680 ± 0.0011	0.8638 ± 0.0001	0.6649 ± 0.0041
G-KD-LoRA UF GPT-2 (LLaMA)	8.14B / 140M	0.2466 ± 0.0004	0.4598 ± 0.0006	0.8637 ± 0.0001	0.6988 ± 0.0018

Table 9: Humor generation performance across various model configurations evaluated using BLEU, ROUGE-L, BERTScore, and humor accuracy. All models are run five times and results are reported as mean $\pm$ standard deviation. F stands for the non harmful dataset, while UF stands for the unfiltered dataset that might generate harmful jokes.

Model	BLEU	ROUGE-L	BERTScore
LoRA F Mistral	0.3328 ± 0.0174	0.4677 ± 0.0222	0.8841 ± 0.0027
LoRA F Qwen	0.2354 ± 0.0008	0.4411 ± 0.0009	0.8442 ± 0.0003
LoRA F LLaMA	0.2381 ± 0.0007	0.4301 ± 0.0011	0.8517 ± 0.0002
LoRA UF Mistral	0.3496 ± 0.0006	0.4730 ± 0.0013	0.8829 ± 0.0005
LoRA UF Qwen	0.2353 ± 0.0008	0.4493 ± 0.0007	0.8393 ± 0.0002
LoRA UF LLaMA	0.2482 ± 0.0003	0.4325 ± 0.0009	0.8486 ± 0.0001
ZS Mistral	0.3083 ± 0.0172	0.4324 ± 0.0260	0.8758 ± 0.0023
ZS Qwen	0.2021 ± 0.0003	0.3356 ± 0.0005	0.8572 ± 0.0003
ZS LLaMA	0.3247 ± 0.0018	0.4285 ± 0.0008	0.8735 ± 0.0003
KD-Critique F GPT-2 (Mistral) (Ravi et al., 2024)	0.3347 ± 0.0008	0.4687 ± 0.0010	0.8806 ± 0.0004
KD-Critique F GPT-2 (Qwen)	0.3393 ± 0.0009	0.4463 ± 0.0012	0.8758 ± 0.0003
KD-Critique F GPT-2 (LLaMA)	0.3361 ± 0.0020	0.4519 ± 0.0012	0.8738 ± 0.0005
KD-Critique UF GPT-2 (Mistral) (Ravi et al., 2024)	0.2115 ± 0.0001	0.3292 ± 0.0001	0.8362 ± 0.0001
KD-Critique UF GPT-2 (Qwen)	0.3393 ± 0.0006	0.4354 ± 0.0007	0.8738 ± 0.0004
KD-Critique UF GPT-2 (LLaMA)	0.3397 ± 0.0014	0.4526 ± 0.0008	0.8746 ± 0.0003
KD-Imitate ZS F GPT-2 (Mistral) (Ravi et al., 2024)	0.3291 ± 0.0010	0.4797 ± 0.0005	0.8843 ± 0.0003
KD-Imitate ZS F GPT-2 (Qwen)	0.3390 ± 0.0007	0.4476 ± 0.0006	0.8769 ± 0.0002
KD-Imitate ZS F GPT-2 (LLaMA)	0.3390 ± 0.0007	0.4476 ± 0.0006	0.8769 ± 0.0002
KD-Imitate ZS UF GPT-2 (Mistral) (Ravi et al., 2024)	0.2121 ± 0.0001	0.3290 ± 0.0001	0.8365 ± 0.0001
KD-Imitate ZS UF GPT-2 (Qwen)	0.3377 ± 0.0004	0.4489 ± 0.0009	0.8773 ± 0.0004
KD-Imitate ZS UF GPT-2 (LLaMA)	0.3355 ± 0.0008	0.4638 ± 0.0009	0.8776 ± 0.0006
KD-Imitate LoRA F GPT-2 (Mistral) (Ravi et al., 2024)	0.3140 ± 0.0007	0.4430 ± 0.0005	0.8828 ± 0.0001
KD-Imitate LoRA F GPT-2 (Qwen)	0.2601 ± 0.0010	0.4504 ± 0.0006	0.8520 ± 0.0002
KD-Imitate LoRA F GPT-2 (LLaMA)	0.2542 ± 0.0004	0.4655 ± 0.0014	0.8456 ± 0.0001
KD-Imitate LoRA UF GPT-2 (Mistral) (Ravi et al., 2024)	0.2112 ± 0.0002	0.3259 ± 0.0001	0.8380 ± 0.0001
KD-Imitate LoRA UF GPT-2 (Qwen)	0.2481 ± 0.0005	0.4668 ± 0.0005	0.8526 ± 0.0002
KD-Imitate LoRA UF GPT-2 (LLaMA)	0.2446 ± 0.0005	0.4515 ± 0.0015	0.8536 ± 0.0001
G-KD-Frozen F GPT-2 (Mistral)	0.4852 ± 0.0024	0.5995 ± 0.0025	0.9111 ± 0.0006
G-KD-Frozen F GPT-2 (Qwen)	0.3811 ± 0.0014	0.4937 ± 0.0017	0.8921 ± 0.0002
G-KD-Frozen F GPT-2 (LLaMA)	0.5196 ± 0.0032	0.6282 ± 0.0023	0.9189 ± 0.0003
G-KD-Frozen UF GPT-2 (Mistral)	0.2736 ± 0.0006	0.4212 ± 0.0003	0.8703 ± 0.0001
G-KD-Frozen UF GPT-2 (Qwen)	0.3200 ± 0.0012	0.4663 ± 0.0005	0.8777 ± 0.0001
G-KD-Frozen UF GPT-2 (LLaMA)	0.2981 ± 0.0005	0.4576 ± 0.0003	0.8736 ± 0.0002
G-KD-LoRA F GPT-2 (Mistral)	0.4024 ± 0.0009	0.5361 ± 0.0010	0.8970 ± 0.0003
G-KD-LoRA F GPT-2 (Qwen)	0.4197 ± 0.0014	0.5863 ± 0.0026	0.9069 ± 0.0006
G-KD-LoRA F GPT-2 (LLaMA)	0.4449 ± 0.0009	0.5661 ± 0.0006	0.9005 ± 0.0002
G-KD-LoRA UF GPT-2 (Mistral)	0.4024 ± 0.0009	0.5361 ± 0.0010	0.8970 ± 0.0003
G-KD-LoRA UF GPT-2 (Qwen)	0.2919 ± 0.0011	0.4410 ± 0.0005	0.8721 ± 0.0002
G-KD-LoRA UF GPT-2 (LLaMA)	0.2981 ± 0.0005	0.4576 ± 0.0003	0.8736 ± 0.0002

Table 10: Automatic evaluation of diversity (mean $\pm$ std) using BLEU, ROUGE-L, and BERTScore against **FT-GPT-2**. F stands for the non harmful dataset, while UF stands for the unfiltered dataset that might generate harmful jokes.

Model	BLEU	ROUGE-L	BERTScore
FT F GPT-2	0.3494 ± 0.0005	0.4730 ± 0.0012	0.8829 ± 0.0005
FT F Qwen	0.2423 ± 0.0006	0.4705 ± 0.0006	0.8464 ± 0.0002
FT F LLaMA	0.2462 ± 0.0007	0.4608 ± 0.0005	0.8545 ± 0.0002
FT UF Qwen	0.2411 ± 0.0009	0.4763 ± 0.0006	0.8402 ± 0.0002
FT UF LLaMA	0.2575 ± 0.0005	0.4664 ± 0.0001	0.8505 ± 0.0002
ZS Qwen	0.2030 ± 0.0003	0.3392 ± 0.0006	0.8642 ± 0.0002
ZS LLaMA	0.3270 ± 0.0019	0.4421 ± 0.0012	0.8804 ± 0.0002
ZS Mistral	0.3264 ± 0.0349	0.4507 ± 0.0319	0.8831 ± 0.0033
KD-Critique F GPT-2 (Mistral) (Ravi et al., 2024)	0.3623 ± 0.0014	0.4756 ± 0.0011	0.8853 ± 0.0002
KD-Critique F GPT-2 (Qwen)	0.3503 ± 0.0010	0.4510 ± 0.0011	0.8806 ± 0.0002
KD-Critique F GPT-2 (LLaMA)	0.3267 ± 0.0016	0.4535 ± 0.0010	0.8773 ± 0.0002
KD-Critique UF GPT-2 (Mistral) (Ravi et al., 2024)	0.2104 ± 0.0001	0.3223 ± 0.0002	0.8557 ± 0.0000
KD-Critique UF GPT-2 (Qwen)	0.3335 ± 0.0008	0.4407 ± 0.0014	0.8786 ± 0.0003
KD-Critique UF GPT-2 (LLaMA)	0.3397 ± 0.0016	0.4568 ± 0.0006	0.8787 ± 0.0002
KD-Imitate ZS F GPT-2 (Mistral) (Ravi et al., 2024)	0.3588 ± 0.0014	0.4881 ± 0.0011	0.8890 ± 0.0003
KD-Imitate ZS F GPT-2 (Qwen)	0.3488 ± 0.0016	0.4531 ± 0.0015	0.8819 ± 0.0003
KD-Imitate ZS F GPT-2 (LLaMA)	0.3488 ± 0.0016	0.4531 ± 0.0015	0.8819 ± 0.0003
KD-Imitate ZS UF GPT-2 (Mistral) (Ravi et al., 2024)	0.2144 ± 0.0002	0.3267 ± 0.0002	0.8609 ± 0.0000
KD-Imitate ZS UF GPT-2 (Qwen)	0.3564 ± 0.0010	0.4548 ± 0.0008	0.8823 ± 0.0003
KD-Imitate ZS UF GPT-2 (LLaMA)	0.3605 ± 0.0009	0.4710 ± 0.0012	0.8820 ± 0.0003
KD-Imitate LoRA F GPT-2 (Mistral) (Ravi et al., 2024)	0.2865 ± 0.0003	0.4363 ± 0.0007	0.8836 ± 0.0002
KD-Imitate LoRA F GPT-2 (Qwen)	0.2334 ± 0.0007	0.4276 ± 0.0005	0.8429 ± 0.0003
KD-Imitate LoRA F GPT-2 (LLaMA)	0.2264 ± 0.0002	0.4393 ± 0.0011	0.8360 ± 0.0002
KD-Imitate LoRA UF GPT-2 (Mistral) (Ravi et al., 2024)	0.2204 ± 0.0001	0.3278 ± 0.0002	0.8583 ± 0.0000
KD-Imitate LoRA UF GPT-2 (Qwen)	0.2201 ± 0.0005	0.4401 ± 0.0017	0.8436 ± 0.0002
KD-Imitate LoRA UF GPT-2 (LLaMA)	0.2220 ± 0.0003	0.4326 ± 0.0009	0.8461 ± 0.0001
G-KD-Frozen F GPT-2 (Mistral)	0.3420 ± 0.0005	0.4713 ± 0.0009	0.8797 ± 0.0003
G-KD-Frozen F GPT-2 (Qwen)	0.3507 ± 0.0003	0.4733 ± 0.0008	0.8828 ± 0.0003
G-KD-Frozen F GPT-2 (LLaMA)	0.3469 ± 0.0015	0.4721 ± 0.0011	0.8832 ± 0.0002
G-KD-Frozen UF GPT-2 (Mistral)	0.2487 ± 0.0003	0.3498 ± 0.0003	0.8398 ± 0.0000
G-KD-Frozen UF GPT-2 (Qwen)	0.2815 ± 0.0009	0.4377 ± 0.0009	0.8676 ± 0.0002
G-KD-Frozen UF GPT-2 (LLaMA)	0.2650 ± 0.0004	0.4340 ± 0.0007	0.8648 ± 0.0001
G-KD-LoRA F GPT-2 (Mistral)	0.3728 ± 0.0018	0.4936 ± 0.0016	0.8821 ± 0.0004
G-KD-LoRA F GPT-2 (Qwen)	0.3751 ± 0.0014	0.5139 ± 0.0009	0.8907 ± 0.0003
G-KD-LoRA F GPT-2 (LLaMA)	0.3602 ± 0.0011	0.4849 ± 0.0008	0.8786 ± 0.0003
G-KD-LoRA UF GPT-2 (Mistral)	0.2468 ± 0.0002	0.3462 ± 0.0001	0.8379 ± 0.0001
G-KD-LoRA UF GPT-2 (Qwen)	0.2790 ± 0.0012	0.4404 ± 0.0009	0.8652 ± 0.0002
G-KD-LoRA UF GPT-2 (LLaMA)	0.2650 ± 0.0004	0.4340 ± 0.0007	0.8648 ± 0.0001

Table 11: Automatic evaluation of diversity (mean $\pm$ std) using BLEU, ROUGE-L, and BERTScore against **LoRA Mistral-7B**. F stands for the non harmful dataset, while UF stands for the unfiltered dataset that might generate harmful jokes.

Model	Win Rate
FT F GPT-2	0.4752
LoRA F Mistral	0.4115
LoRA F Qwen	0.5206
LoRA F LLaMA	0.5786
FT UF GPT-2	0.5264
LoRA UF Mistral	0.7459
LoRA UF Qwen	0.5246
LoRA UF LLaMA	0.5489
ZS Mistral	0.7536
ZS LLaMA	0.4056
ZS Qwen	0.5359
KD-Critique F GPT-2 (Mistral)	0.4777
KD-Critique F GPT-2 (Qwen)	0.4611
KD-Critique F GPT-2 (LLaMA)	0.4369
KD-Critique UF GPT-2 (Mistral)	0.4872
KD-Critique UF GPT-2 (Qwen)	0.4897
KD-Critique UF GPT-2 (LLaMA)	0.4509
KD-Imitate ZS F GPT-2 (Mistral)	0.4778
KD-Imitate ZS F GPT-2 (Qwen)	0.4610
KD-Imitate ZS F GPT-2 (LLaMA)	0.4721
KD-Imitate ZS UF GPT-2 (Mistral)	0.4752
KD-Imitate ZS UF GPT-2 (Qwen)	0.4761
KD-Imitate ZS UF GPT-2 (LLaMA)	0.4521
KD-Imitate LoRA F GPT-2 (Mistral)	0.5127
KD-Imitate LoRA F GPT-2 (Qwen)	0.2561
KD-Imitate LoRA F GPT-2 (LLaMA)	0.2640
KD-Imitate LoRA UF GPT-2 (Mistral)	0.1601
KD-Imitate LoRA UF GPT-2 (Qwen)	0.2835
KD-Imitate LoRA UF GPT-2 (LLaMA)	0.2743
G-KD-Frozen F GPT-2 (Mistral)	0.5938
G-KD-Frozen F GPT-2 (Qwen)	0.6032
G-KD-Frozen F GPT-2 (LLaMA)	0.5984
G-KD-Frozen UF GPT-2 (Mistral)	0.5689
G-KD-Frozen UF GPT-2 (Qwen)	0.6125
G-KD-Frozen UF GPT-2 (LLaMA)	0.6314
G-KD-LoRA F GPT-2 (Mistral)	0.5629
G-KD-LoRA F GPT-2 (Qwen)	0.5901
G-KD-LoRA F GPT-2 (LLaMA)	0.6063
G-KD-LoRA UF GPT-2 (Mistral)	0.5464
G-KD-LoRA UF GPT-2 (Qwen)	0.5649
G-KD-LoRA UF GPT-2 (LLaMA)	0.6372

Table 12: LLM-as-Judge win rate across different model configurations. Higher values indicate stronger preference by the judge model when comparing generated punchlines. F stands for the non harmful dataset, while UF stands for the unfiltered dataset that might generate harmful jokes.

Setup	Punchline	Setup	Punchline
What did the buddhist say to the hot dog vendor?	He said, "Make me one with everything."	How can dogs float in water?	Because they're good buoys.
FT GPT-2	Make me one with everything.	**FT GPT-2**	They have two left feet.
LoRA Mistral	What did the buddhist say to the hot dog vendor? Make me one with everything.	**LoRA Mistral**	They use life preservers.
KD-Critique GPT-2 (Qwen)	May you have a heavenly experience, but at least you can get some peace out of it!	**KD-Critique GPT-2 (Qwen)**	They use their tail feathers for buoyancy!
KD-Imitate ZS GPT-2 (Mistral)	You're really dipping your toes in the holy water.	**KD-Imitate ZS GPT-2 (Mistral)**	They have a "sea-tastic" sense of smell!
KD-Imitate LoRA GPT-2 (Mistral)	I'll have a bun with everything, please.	**KD-Imitate LoRA GPT-2 (Mistral)**	They have two hooves.
G-KD-Frozen GPT-2 (Mistral)	**I'm feeling a little saucy.**	**G-KD-Frozen GPT-2 (Mistral)**	**They're good at bark-ing!**
G-KD-LoRA GPT-2 (Mistral)	Bud, bud. I don't wanna buy that hot dog. I'm buddha!	**G-KD-LoRA GPT-2 (Mistral)**	They can't. The beer is just holding them back.
Why was the peach late to work?	He had to make a pit stop on the way there.	Why did the cookie get fired from his job?	He came to work baked.
FT GPT-2	Because it was in the Peach Jammer.	**FT GPT-2**	He didn't show up!
LoRA Mistral	Because he was pruned!	**LoRA Mistral**	Because he was a little doughy.
KD-Critique GPT-2 (Qwen)	It got lost in the marshmallow maze!	**KD-Critique GPT-2 (Qwen)**	Because he couldn't crack under the pressure.
KD-Imitate ZS GPT-2 (Mistral)	Because it was worried about getting peaches-terated.	**KD-Imitate ZS GPT-2 (Mistral)**	Because he couldn't stop eating the batter.
KD-Imitate LoRA GPT-2 (Mistral)	Because it was in the can.	**KD-Imitate LoRA GPT-2 (Mistral)**	He didn't get a promotion.
G-KD-Frozen GPT-2 (Mistral)	**Because it was out of juice!**	**G-KD-Frozen GPT-2 (Mistral)**	**Because he was too cookie-cited!**
G-KD-LoRA GPT-2 (Mistral)	Because it overswept.	**G-KD-LoRA GPT-2 (Mistral)**	He didn't cut the super cheezit down.

Table 13: Four examples of generated jokes for different model configurations (reordered).