Teaching with Lies: Curriculum DPO on Synthetic Negatives for Hallucination Detection

Anonymous ACL submission

Abstract

Aligning large language models (LLMs) to accurately detect hallucinations remains a significant challenge due to the sophisticated nature of hallucinated text. Recognizing that hallucinated samples typically exhibit higher deceptive quality than traditional negative samples, we use these carefully engineered hallucinations as negative examples in the DPO alignment procedure. Our method incorporates a curriculum learning strategy, gradually transitioning the training from easier samples, identified based on the greatest reduction in probability scores from independent fact checking models, to progressively harder ones. This structured difficulty scaling ensures stable and incremental learning. Experimental evaluation demonstrates that our HaluCheck models, trained with curriculum DPO approach and high quality negative samples, significantly improves model performance across various metrics, achieving improvements of upto 24% on difficult benchmarks like MedHallu and HaluEval. Additionally, HaluCheck models demonstrate robustness in zero-shot settings, significantly outperforming larger state-of-the-art models across various benchmarks.

1 Introduction

Large language models (LLMs) have achieved impressive performance across numerous NLP tasks, yet their deployment is limited by a tendency to produce fluent but factually incorrect “hallucinations.” Such errors erode trust and carry serious risks in domains with LLM applications like healthcare (Singhal et al., 2022), software-development (Krishna et al., 2024) and Law (Lai et al., 2024). Although various detection and mitigation strategies often based on external fact-checkers or simplistic negative samples have been proposed, they struggle to identify sophisticated, plausibly crafted falsehoods.

To address these challenges, we introduce a novel alignment strategy leveraging Direct Preference Optimization (DPO) (Rafailov et al., 2023), enhanced through a curriculum learning (Bengio et al., 2009a) (Elman, 1993a) approach specifically tailored for hallucination detection. Our approach incorporates high quality hallucinated samples as negative samples into the alignment process instead of the usual low quality negative samples that are often selected from failed generations.

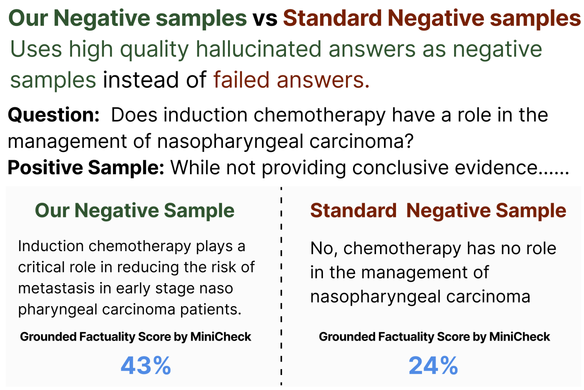

Figure 1: Illustration of the qualitative difference between standard negative samples used in conventional DPO alignment and our proposed method, which leverages carefully curated hallucinated answers as high-quality negative examples in DPO alignment.

We introduce **HaluCheck**, a family of Hallucination detection LLMs at two scales aligned via our curriculum-based DPO framework. We conduct extensive evaluations on the MedHallu (Pandit et al., 2025) and HaluEval (Li et al., 2023) benchmarks and zero-shot evaluation on DROP, CovidQA, and PubMedQA, demonstrating that HaluCheck substantially outperforms existing baselines, including the widely adopted Llama-3.2 (1B and 3B) models. Notably, HaluCheck 3B yields up to a 24% relative gain across core detection metrics (accuracy, precision, recall, and F1-score), while remaining competitive with far larger models such as GPT-4o. Our contributions are summarized as follows:

1. We introduce a novel curriculum based sam-

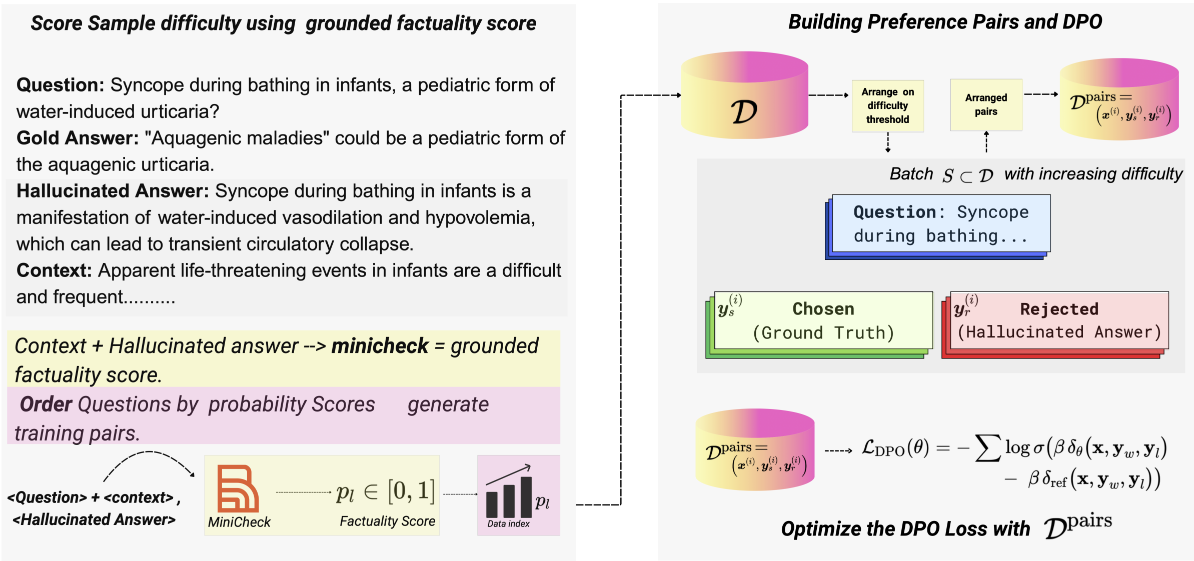

Figure 2: Figure showing the pipeline for selecting high-quality hallucinated negatives for Direct Preference Optimization (DPO). Each question and context is paired with a hallucinated answer and scored for grounded factuality via MiniCheck, then ranked by difficulty. In each batch, gold references (chosen) and top-ranked hallucinations (rejected) form preference pairs. These pairs optimize the DPO objective, ensuring training against vetted, high-quality negatives rather than arbitrary failures.

pling strategy that progressively selects hallucinated samples of increasing difficulty ranges obtained from fact verification models to enhance alignment training.

2. We introduce **HaluCheck**, a suite of 1B–3B parameter models aligned with our DPO curriculum that leverages high-quality negative samples to deliver hallucination detection gains outperforming state of the art LLMs.

3. Results demonstrate strong transferability of HaluCheck across multiple benchmarks and domains (Sec. 5), including zeroshot evaluation, confirming robustness in hallucinations detection task on diverse datasets.

2 Related Works

Finetuning Models for Hallucination Detection Recent research shows that both model-centric finetuning and sampling-based methods effectively detect hallucinations. LYNX (Ravi et al., 2024), an open-source detector refined with distilled chain-of-thought reasoning, outperforms closed-source alternatives and provides HaluBench(Ravi et al., 2024), a diverse benchmark of semantically perturbed hallucinations. FACTCHECKMATE (Alnuhait et al., 2024) preemptively flags hallucination risks via a lightweight MLP on hidden states and uses an intervention network to boost factuality with minimal overhead. SelfCheckGPT (Manakul et al., 2023) requires no output probabilities or external knowledge: it samples multiple outputs and applies consistency measures such as BERTScore (Zhang et al., 2019a) at both sentence and passage levels. Existing work does not exploit alignment methods such as DPO (Rafailov et al., 2023), despite their proven effectiveness. We introduce the first DPO approach that leverages curated hallucinated negatives, markedly improving hallucination detection.

Hallucination Detection Task Hallucination in large language models (LLMs) has been extensively documented across various natural language processing tasks, such as machine translation (Lee et al., 2019), dialogue systems (Balakrishnan et al., 2019), text summarization (Durmus et al., 2020), and question answering (Sellam et al., 2020), as detailed in recent survey literature (Ji et al., 2023). Benchmarks like Hades (Liu et al., 2022) and HaluEval (Li et al., 2023) offer strong hallucination-detection protocols, and MedHallu (Pandit et al., 2025) provides carefully crafted adversarial answers that are ideal for our alignment approach.

Model	Average F1	MedHallu (Pandit et al., 2025)			HaluEval (Li et al., 2023)		
		F1	Precision	Accuracy	F1	Precision	Accuracy
Qwen-2.5 1.5B	0.464	0.227	0.642	0.525	0.701	0.568	0.610
LLama-3.2 1B	0.237	0.108	0.406	0.494	0.366	0.450	0.466
Qwen-2.5 3B	0.638	0.606	0.495	0.492	0.671	0.506	0.512
LLama-3.2 3B	0.612	0.499	0.696	0.566	0.726	0.743	0.732
LLama-3.1 8B	0.571	0.522	0.791	0.608	0.620	**0.903**	0.711
Qwen-2.5 14B	0.720	0.619	0.691	0.633	0.821	0.862	0.829
GPT 4o	**0.799**	0.737	0.723	0.772	**0.862**	**0.896**	**0.867**
HalluCheck-Llama 1B	0.637	0.664	0.511	0.527	0.611	0.481	0.468
HalluCheck-Llama 3B	0.756	**0.759**	**0.845**	**0.782**	0.753	0.857	0.767

Table 1: Performance comparison of various models on the MedHallu and HaluEval hallucination detection benchmarks. Our proposed HaluCheck variants (1B and 3B) consistently outperform significantly larger foundational models. Notably, HaluCheck 3B demonstrates superior or comparable performance across both benchmarks, highlighting its efficiency and effectiveness despite its smaller size. Best scores are **bold**, runners-up are underlined.

For the purpose of this work we choose MedHallu and HaluEval for the DPO alignment, as they have high quality hallucinated samples. Our proposed method is agnostic of task, and can be extended to other hallucination detection tasks like in summarization and dialogue answering setting.

3 Hallucination Detection and Alignment

Problem formulation For each sample i we define Let $\boldsymbol{x}^{(i)}$ denote the detection prompt (context + question + task instruction), $\boldsymbol{y}^{(i)}_{\text{hall}}$ represent the *hallucinated* class completion, and $\boldsymbol{y}^{(i)}_{\text{true}}$ represent the *factual* class completion. We define $l^{(i)} \in \{0,1\}$ as the gold label, where a value of 1 indicates hallucination. From every labelled example we obtain a **preference pair** $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}_w, \boldsymbol{y}^{(i)}_l)$, where

$$(\boldsymbol{y}^{(i)}_w, \boldsymbol{y}^{(i)}_l) = \left\{ (\boldsymbol{y}^{(i)}_{\text{true}}, \boldsymbol{y}^{(i)}_{\text{hall}}) \right.$$

MiniCheck-Based Grounding Difficulty scoring Before curriculum partitioning, we evaluate how well each hallucinated output is supported by its context using MiniCheck (Tang et al., 2024). For each example $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}_{\text{hall}})$, we treat question $=$ $\boldsymbol{y}^{(i)}_{\text{hall}}$ and context $= \boldsymbol{x}^{(i)}$, and compute the grounding probability

$$p_l^{(i)} = \mathcal{F}\big(\text{question} = \boldsymbol{y}^{(i)}_{\text{hall}} \;\big|\; \text{context} = \boldsymbol{x}^{(i)}\big).$$

We then use $p_l^{(i)}$ to score difficulty and drive our curriculum stages. After sorting all examples by $p_l^{(i)}$ (ascending), $\{\mathcal{B}_s\}_{s=1}^{S} \leftarrow$ split into S bins. Lower p_l indicates easier hallucination cases, ensuring the curriculum starts with easy (high-grounding) and gradually moves to harder ones.

DPO Objective for Hallucination Detection Let π_θ be the current policy and π_{ref} the frozen reference model. With trust–region parameter β, and $\sigma(z) = 1/(1+e^{-z})$ the batch loss is:

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\theta) = -\sum_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\in\mathcal{B}} \log \sigma\Big(\beta\big[\log \pi_\theta(\boldsymbol{y}_w \,|\, \boldsymbol{x}) \\ - \log \pi_\theta(\boldsymbol{y}_l \,|\, \boldsymbol{x})\big] - \beta\big[\log \pi_{\text{ref}}(\boldsymbol{y}_w \,|\, \boldsymbol{x}) \\ - \log \pi_{\text{ref}}(\boldsymbol{y}_l \,|\, \boldsymbol{x})\big]\Big). \qquad (1)\end{aligned}$$

We provide a detailed algorithm for this pipeline in the supplementary (Alg. 1)

4 Experimental Setup

We describe the setup in the following section, and have a detailed section in supplementary C and D

Model & Datasets We fine-tune `Llama-3.2` backbones (1 B and 3 B parameters) with LoRA adapters under the Direct Preference Optimization objective, using a joint corpus drawn from MedHallu and HaluEval. Hallucination detection is cast as binary classification via task-specific prompts.

Sampling Strategy & Curriculum Learning Negative examples are high-quality hallucinations scored by the MiniCheck fact-verifier. We sort them by decreasing MiniCheck confidence drop and train with a curriculum that proceeds from the easiest to the hardest negatives, yielding smoother and more robust convergence.

5 Results

In the upcoming sections, 5.1 ❶ we demonstrate that our HaluCheck models (1B and 3B) significantly outperform foundation LLMs despite their

Model	DROP	CovidQA	PQA	Avg
Llama 3.2 3B	52.50	56.10	55.20	54.60
HaluCheck 3B	**57.30**	**62.50**	57.70	**59.16**
GPT-3.5-Turbo	57.20	56.70	**62.80**	58.90

Table 2: Accuracy (%) on DROP, CovidQA and PQA (PubMedQA) for the baseline Llama 3.2 3B, our HaluCheck 3B, and GPT-3.5-Turbo (results from HaluBench (Ravi et al., 2024)). Results indicate strong performance of HaluCheck in zeroshot setting.

Sample Type	Easy		Medium		Hard	
	Mean	Median	Mean	Median	Mean	Median
Standard Negative	0.282	0.202	0.273	0.201	0.248	0.182
Our Hallucinated	0.303	0.202	0.379	0.269	0.391	0.294

Table 3: Grounded factuality scores (MiniCheck `true_prob`; higher is harder to spot) for standard negatives versus our curated hallucinated negatives, averaged over difficulty tiers for MedHallu dataset. The curated set provides consistently higher means and medians, confirming its superiority as training negatives for DPO.

smaller size. In Sec.5.2, we further show that ❷ HaluCheck generalizes effectively to unseen datasets in a zero-shot setting, clearly outperforming its baseline model. In Sec.5.3, we validate the importance of using curated hallucinated samples rather than standard failed generations as negatives in DPO, showing that ❸ our model trained with curated hallucinated answers as negatives achieves superior performance. Finally, in Sec. A.1 and A.2, we conduct ablations demonstrating HaluCheck's superior transferable skills when trained on individual datasets, and highlight the benefits of curriculum-based sampling over random selection.

5.1 HaluCheck vs Baseline

As presented in Table 1 **HaluCheck 3B**, trained with DPO on hallucinated answers as high quality negative samples, significantly outperforms similar and larger sized models. On HaluEval, it achieves an F1-score of `0.753`, surpassing the baseline `LLama-3.2 3B` (F1: `0.726`). On MedHallu, it outperforms the base model by `+26%` F1 gain. Similarly, **HaluCheck 1B** shows strong performance on MedHallu (F1: `0.711`), while baseline LLama-3.2 1B lags behind (F1: `0.366`). ❶ These results highlight our curriculum-based DPO approach's efficacy in enhancing hallucination detection while maintaining computational efficiency.

5.2 Zero-shot evaluation

To gauge out-of-domain robustness, we ran a strict zeroshot test of **HaluCheck 3B** without any extra tuning or prompt changes against the backbone model `Llama-3.2 3B` and much larger `GPT-3.5-Turbo` on three external QA style hallucination benchmarks taken from the HaluBench dataset (Ravi et al., 2024): DROP (Dua et al., 2019), CovidQA (Möller et al., 2020), and PubMedQA (Jin et al., 2019). As shown in Table 2, HaluCheck 3B outperforms the Llama 3.2 3B model across the board, improving accuracy by **+4.8%**, **+6.4%**, and **+2.5%** on the respective datasets, and also outperforming the `GPT-3.5-Turbo` on CovidQA by a substantial margin. ❷ These consistent gains achieved affirm that our curriculum based DPO alignment with using hallucinated samples as a high quality negative samples confers transferable hallucination detection skills that scale to unseen datasets.

5.3 DPO using Hallucinated vs Standard negative samples

We show the importance of choosing curated hallucinated answers as a negative sample for DPO alignment by comparing the performance of Llama-3.2 3B model trained with standard negative samples. We sample these standard negative samples, by querying LLM for the question, and keeping the failed answers as negative samples, that is generally chosen as negative samples for DPO. We report the results in Table 7, which clearly indicates that ❸ HaluCheck outperforms the later trained model. Also, to further back this choice, we report the grounded factuality score for the hallucinated answers from MedHallu and the standard negative samples we created, in Table 3, showing the superiority of the samples as negatives for DPO, thereby being a better choice for DPO.

6 Conclusion

We present **HaluCheck** a curriculum-guided Direct Preference Optimization (DPO) framework for training an LLM for reliable hallucination detection task. A key contribution lies in replacing generic, model-generated failures with carefully curated, difficulty-ranked hallucinated samples as negative preferences during DPO alignment. This structured curriculum yields consistent gains, outperforming larger state-of-the-art models on multiple benchmarks and zero-shot tasks. Ablation results further validate that difficulty-aware negative sampling markedly strengthens the robustness of smaller language models.

Limitations

Our proposed approach, while effective, exhibits certain limitations worth acknowledging. The curriculum-based Direct Preference Optimization (DPO) heavily relies on the quality and accuracy of the external fact-verification model (MiniCheck), potentially propagating any inherent biases or inaccuracies into our training process. Furthermore, our evaluations primarily focus on hallucinations within question-answering contexts, leaving unexplored the effectiveness in other NLP tasks such as dialogue generation, summarization, or multilingual settings. Additionally, treating hallucination detection purely as a binary classification task restricts the model's ability to identify partial or span-level hallucinations, thus limiting fine-grained interpretability. Lastly, although zeroshot evaluations suggest good generalization, there remains a risk of overfitting to dataset-specific adversarial patterns used during training, which may affect broader applicability and robustness.

Ethics statement

Our work develops **HaluCheck** to improve reliable detection of hallucinations in LLM outputs, with the goal of reducing the risk of disseminating misleading or harmful information. Our work uses publicly available MedHallu, and HaluEval data under MIT licenses We acknowledge that our reliance on an external fact-verification model may introduce its own biases, and users should avoid treating automated detectors as infallible; human oversight remains essential, especially in high-stakes domains like healthcare or law. We encourage ongoing evaluation for fairness and transparency, and recommend that practitioners combine our approach with diverse verification methods to mitigate unintended biases or misuse.

References

Deema Alnuhait, Neeraja Kirtane, Muhammad Khalifa, and Hao Peng. 2024. Factcheckmate: Preemptively detecting and mitigating hallucinations in lms. *arXiv preprint arXiv:2410.02899*.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Florence, Italy. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009a. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009b. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Monojit Choudhury, Kalika Bali, Sunayana Sitaram, and Ashutosh Baheti. 2017. Curriculum design for code-switching: Experiments with language identification and language modeling with deep neural networks. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 65–74.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Jeffrey L Elman. 1993a. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Jeffrey L Elman. 1993b. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Madhava Krishna, Bhagesh Gaur, Arsh Verma, and Pankaj Jalote. 2024. Using llms in software requirements specifications: An empirical evaluation. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 475–483.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S Yu. 2024. Large language models in law: A survey. *AI Open*.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in neural machine translation.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *Preprint*, arXiv:2305.11747.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. *Preprint*, arXiv:2104.08704.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. Covid-qa: A question answering dataset for covid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Shrey Pandit, Jiawei Xu, Junyuan Hong, Zhangyang Wang, Tianlong Chen, Kaidi Xu, and Ying Ding. 2025. Medhallu: A comprehensive benchmark for detecting medical hallucinations in large language models. *Preprint*, arXiv:2502.14302.

Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Curry-dpo: Enhancing alignment using curriculum learning & ranked preferences. *arXiv preprint arXiv:2403.07230*.

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. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. 2024. Lynx: An open source hallucination evaluation model. *arXiv preprint arXiv:2407.08488*.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 453–463.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation. *Preprint*, arXiv:2004.04696.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2022. Large language models encode clinical knowledge. *arXiv preprint arXiv:2212.13138*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *arXiv preprint arXiv:2404.10774*.

Yi Tay, Shuohang Wang, Luu Anh Tuan, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. *arXiv preprint arXiv:1905.10847*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6095–6104.

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

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019b. Curriculum learning for domain adaptation in neural machine translation. *arXiv preprint arXiv:1905.05816*.

A Ablations

A.1 Training on individual datasets

Only Train on `MedHallu` When we fine-tune the `HaluCheck-Llama-3B` detector exclusively on the `MedHallu` DPO set, the model achieves strong in-domain performance, with an `F1` of `0.729`, precision of `0.892`, and accuracy of `0.784` on the `MedHallu` benchmark. However, this specialization comes at the expense of generalization: when evaluated on `HaluEval`, the same model's `F1` drops to `0.627`, precision to `0.578`, and accuracy to `0.593`. These results demonstrate that training solely on one dataset leads to overfitting to its particular style and content, limiting cross-dataset transfer.

Only Train on `HaluEval` Conversely, training exclusively on the `HaluEval` DPO set yields a model that excels on `HaluEval` (`F1 = 0.793`, precision = `0.794`, accuracy = `0.793`), but underperforms on `MedHallu` (`F1 = 0.675`, precision = `0.623`, accuracy = `0.644`). Although the in-domain metrics on `HaluEval` are highest among the single-dataset trainings, the drop in `MedHallu` performance again highlights the narrow adaptation of the model to the peculiarities of its training set.

Training on each dataset in isolation yields high in-domain accuracy but poor transfer. In contrast, combining both DPO sets produces a model that maintains strong performance across `MedHallu` and `HaluEval`, underscoring the importance of diverse hallucination examples for robust detector alignment.

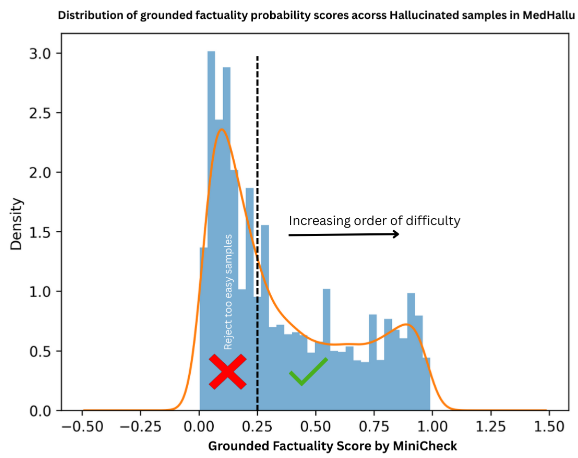

Figure 3: Figure showing the grounded factuality of the hallucinated samples from MedHallu dataset. We keep only the samples that have a score above 0.25.

Algorithm 1 Curriculum-Based DPO Alignment for Hallucination Detection

Require: Detection data $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}_{\text{true}}, \boldsymbol{y}^{(i)}_{\text{hall}}, l^{(i)})\}_{i=1}^{N}$, fact-checker $\mathcal{F}$ (using MiniCheck, returns probability), policy π_θ, frozen ref. policy π_{ref}, stages S
Ensure: Fine-tuned detector π_θ

 # Score difficulty
for each $(\boldsymbol{x}, \boldsymbol{y}_{\text{true}}, \boldsymbol{y}_{\text{hall}}, l)$ **do**
  $p_l \leftarrow \mathcal{F}(\boldsymbol{y}_l \mid \boldsymbol{x})$
end for
 # Partition into stages
sort by p_l (asc.) and split into $\{\mathcal{B}_s\}_{s=1}^{S}$
 # Generate preference pairs
for $i = 1, \ldots, N$ **do**
  $\boldsymbol{y}_w^{(i)} \leftarrow \boldsymbol{y}_{\text{true}}^{(i)}$
  $\boldsymbol{y}_l^{(i)} \leftarrow \boldsymbol{y}_{\text{hall}}^{(i)}$
  store $(\boldsymbol{x}^{(i)}, \boldsymbol{y}_w^{(i)}, \boldsymbol{y}_l^{(i)})$
end for
 # Stage-wise DPO fine-tuning
for $s = 1, \ldots, S$ **do**
  Define:
  $\delta_\theta(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) = \log \pi_\theta(\boldsymbol{y}_w \mid \boldsymbol{x}) - \log \pi_\theta(\boldsymbol{y}_l \mid \boldsymbol{x})$
  $\delta_{\text{ref}}(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) = \log \pi_{\text{ref}}(\boldsymbol{y}_w \mid \boldsymbol{x}) - \log \pi_{\text{ref}}(\boldsymbol{y}_l \mid \boldsymbol{x})$
  Minimize over $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) \in \mathcal{B}_s$:
  $$\mathcal{L}_{\text{DPO}}(\theta) = -\sum \log \sigma\big(\beta\, \delta_\theta(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) - \beta\, \delta_{\text{ref}}(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l)\big)$$
end for
return π_θ

A.2 Random vs Curriculum learning DPO

As Table 5 shows, replacing the usual *random* selection of negative samples with a *curriculum* that feeds the model increasingly difficult hallucinations produces a clear performance boost on both benchmarks and at both parameter scales. With just 1 B parameters, curriculum guided DPO lifts F1 on MedHallu 0.528 for the random baseline to 0.664 and on HaluEval from 0.446 to 0.611 gains that transform a lightweight detector from marginal to competitive accuracy. The effect is even more pronounced at 3 B curriculum training drives MedHallu F1 to 0.759 and HaluEval F1 to 0.753, surpassing the random counterpart by a wide margin and closing much of the gap to models an order of magnitude larger. These results confirm the intuition that hard, well vetted negatives presented in a staged fashion teach the model subtler decision boundaries than a grab-bag of arbitrary failures, leading to more robust hallucination detection with no increase in parameter count or compute budget.

B Additional Related Works

Curriculum learning Curriculum learning represents a training paradigm that strategically presents data samples in a meaningful sequence, effec-

Model	DPO set		MedHallu			HaluEval		
	MedHallu	HaluEval	F1	Precision	Accuracy	F1	Precision	Accuracy
HalluCheck-Llama 3B	✓	✗	0.729	0.892	0.784	0.627	0.578	0.593
HalluCheck-Llama 3B	✗	✓	0.675	0.623	0.644	0.793	0.794	0.793
HalluCheck-Llama 3B	✓	✓	0.759	0.845	0.782	0.733	0.857	0.767

Table 4: Performance over training with different train sets.

Model	MedHallu F1	HaluEval F1
HalluCheck 1B (Random)	52.80	44.60
HalluCheck 1B (Curr.)	66.40	61.10
HalluCheck 3B (Random)	69.40	63.10
HalluCheck 3B (Curr.)	75.90	75.30

Table 5: F1 comparison of curriculum-guided vs. random sampling for HalluCheck models on MedHallu and HaluEval.

tively managing and optimizing the information a model encounters at each training step (Elman, 1993b; Bengio et al., 2009b). Research has demonstrated the effectiveness of progressing from simple to complex examples across various NLP tasks, including language modeling (Choudhury et al., 2017; Xu et al., 2020), reading comprehension (Tay et al., 2019), question answering (Sachan and Xing, 2016), and machine translation (Zhang et al., 2019b). In the context of LLM alignment, curriculum learning applications remain limited, with (Pattnaik et al., 2024) applying curriculum learning principles within the DPO framework for alignment.

C Detailed experimental setup

C.1 Model and Dataset Details

We adopt the publicly released `Llama-3.2` checkpoints at two scales (1 B and 3 B parameters). LoRA hyper-parameters follow Hu et al. (2022): rank=8, α=32, dropout=0.05, and target modules `q_proj`, `k_proj`, `v_proj`, and `o_proj`. Training data comprise 9 000 examples from MedHallu's `pqa_artificial` split plus 8 000 items (80 %) from the HaluEval training partition, forming 17 000 DPO preference pairs. Evaluation is conducted on the 1 000-example MedHallu `pqa_labeled` set and the held-out 2 000 HaluEval test items.

C.2 Curriculum Construction

For every hallucinated answer h_i paired with context c_i, the MiniCheck verifier returns a grounding probability p_i. Examples with $p_i < 0.25$ (very poor grounding) are discarded. The remainder are sorted by ascending values of p_i. DPO training proceeds batch wise on the sorted data for four epochs, with all batches trained per epoch, thereby gradually exposing the model to increasingly difficult negatives. Table 6 in the main paper reports ablations over alternative cut-offs; the chosen 0.25–1.0 range yields the highest F1 scores, consistent with the grounded factuality distribution visualized in Figure 3.

D Implementation details

Training was performed using Direct Preference Optimization (DPO) with hyperparameters set as follows: learning rate = 1×10^{-5}, beta = 0.1, gradient accumulation steps = 4, per-device batch size = 4, and total epochs = 25. We used a paged AdamW optimizer with 8-bit quantization and mixed-precision training (FP16) for computational efficiency. Sequential sampling was used during training to maintain curriculum learning order. The model's performance was periodically assessed on the MedHallu labeled validation set. Evaluation metrics included accuracy, precision, recall, and F1-score, computed both overall and separately by difficulty (easy, medium, hard).

E LLMs Used in Discriminative Tasks

GPT-4o and GPT-4o mini. GPT-4o (OpenAI et al., 2024) are a series of commercial LLMs developed by OpenAI. Renowned for their state-of-the-art performance, these models have been extensively utilized in tasks such as medical hallucination detection. Our study employs the official API provided by the OpenAI platform to access these models. For all other models below, we implement them through Hugging Face package.

Llama-3.1 and Llama-3.2. Llama-3.1 and Llama-3.2 (Meta, 2024) are part of Meta's open-source multilingual LLMs, Llama 3.1 (July 2024) includes 8B, 70B, and 405B parameter models optimized for multilingual dialogue. Llama 3.2 (September 2024) offers 1B, 3B, 11B, and 90B models with enhanced accuracy and speed. We use Llama 3.2 1B and 3B models as our backbone for

Split Range	Model	Avg F1	MedHallu			HaluEval		
			F1	Prec	Acc	F1	Prec	Acc
0.00–0.75	HaluCheck 1B	0.499	0.404	0.717	0.596	0.595	0.491	0.458
	HaluCheck 3B	0.714	0.729	0.892	0.784	0.699	0.812	0.728
0.25–1.00	HaluCheck 1B	0.637	0.664	0.511	0.527	0.611	0.481	0.468
	HaluCheck 3B	0.756	0.759	0.845	0.782	0.753	0.857	0.767
0.25–0.75	HaluCheck 1B	0.625	0.651	0.501	0.511	0.599	0.512	0.469
	HaluCheck 3B	0.712	0.696	0.727	0.704	0.728	0.824	0.739
0.00–1.00	HaluCheck 1B	0.614	0.622	0.601	0.459	0.606	0.494	0.455
	HaluCheck 3B	0.743	0.743	0.905	0.770	0.744	0.829	0.759

Table 6: **Ablation over curriculum difficulty cut-offs**. Each split indicates the MiniCheck grounding-probability interval used when selecting hallucinated negatives. "Avg F1" is the mean F1 score across MedHallu and HaluEval; higher is better for all metrics.

Model	F1	Precision	Accuracy
HaluCheck 1B	0.664	0.511	0.527
Llama-3.2 1B-SN	0.622	0.494	0.491
HaluCheck 3B	0.729	0.845	0.782
Llama-3.2 3B-SN	0.691	0.772	0.717

Table 7: **Hallucination detection on the MedHallu dataset**. "SN" models were aligned with standard negative samples in DPO, while HaluCheck models were aligned with curated hallucinated negatives. Higher is better on all metrics.

training DPO, and also use the Llama 3.1 8B model in our evaluation table for performance comparison

Qwen2.5. Qwen2.5 (Team, 2024) is an advanced LLM designed to handle complex language tasks efficiently. It has been applied in various domains, including medical hallucination detection. We use the 3B, 7B and 14B variants in our work.

F Hardware Resources and Computational Costs

During the DPO training process using LoRA, we primarily used the `Llama-3.2 1B` and `Llama-3.2 3B` model as a base model for our HaluCheck Model, running it for 12 hours on an NVIDIA RTX A6000 GPU with 48,685 MiB of RAM. Additionally, we employed models such as `Qwen2.5-1.5B, 3B, 14B`, and GPT models as evaluators for benchmarkings. To enhance the efficiency and speed of our code execution, we utilized software tools like `vLLM` and implemented batching strategies. These optimizations were critical for managing the computational load and ensuring timely processing of our experiments.