ENHANCING LLM ROBUSTNESS TO PERTURBED INSTRUCTIONS: AN EMPIRICAL STUDY

Aryan Agrawal*, **Lisa Alazraki***, **Shahin Honarvar, Marek Rei**
Imperial College London

ABSTRACT

Large Language Models (LLMs) are highly vulnerable to input perturbations, as even a small prompt change may result in a substantially different output. Existing methods to enhance LLM robustness are primarily focused on perturbed data samples, whereas improving resiliency to perturbations of task-level instructions has remained relatively underexplored. In this work, we focus on character- and word-level edits of task-specific instructions, which substantially degrade downstream performance. We experiment with a variety of techniques to enhance the robustness of LLMs, including self-denoising and representation alignment, testing different models (Llama 3 and Flan-T5), datasets (CoLA, QNLI, SST-2) and instructions (both task-oriented and role-oriented). We find that, on average, self-denoising—whether performed by a frozen LLM or a fine-tuned model—achieves substantially higher performance gains than alternative strategies, including more complex baselines such as ensembling and supervised methods. We share our data and code at `https://github.com/ary4n99/llm-robustness`.

1 INTRODUCTION

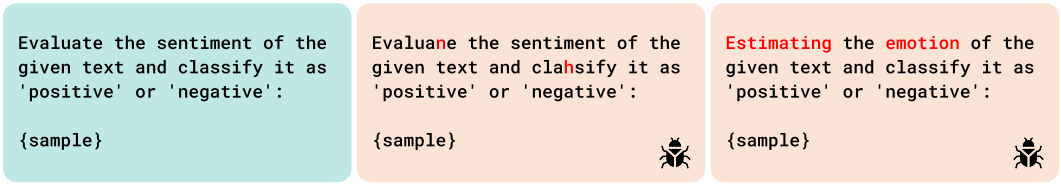
Despite achieving impressive performance in increasingly sophisticated tasks (Nori et al., 2023; Roemmele & Gordon, 2024), LLMs remain sensitive to input perturbations (Wang et al., 2024b). While human performance in natural language tasks is resilient to small alterations in the problem description (Walkington et al., 2019), LLMs have consistently been observed to shift their output dramatically even with minor changes to the input (Moradi & Samwald, 2021; Wang et al., 2022c; Alazraki et al., 2023; Gulati et al., 2024; Honarvar et al., 2025). Prior literature extensively investigates improving LLM robustness when individual data samples are perturbed (Hu et al., 2024; Wang et al., 2024a; Chen et al., 2025). On the other hand, methods for handling perturbations of task-level instructions (i.e., fixed templates that are combined with each data point to help solve a task) are less researched.

Intuitively, instruction quality and readability have a considerable impact on performance: perturbing the instruction can potentially lead the LLM to misunderstand the task and fail on all samples. Indeed, prior work finds that LLM proficiency varies widely when instructions are paraphrased (Mizrahi et al., 2024; Zhu et al., 2024) or individual words are replaced, added or removed (Gu et al., 2023; Zhu et al., 2024). Similarly, Sun et al. (2024) study the performance of instruction-tuned LLMs when test-time instructions are phrased differently from the training data, and observe substantial degradation across different models and tasks. As a solution, they propose aligning the internal model representations of the rephrased instructions to those of the original ones.

In this work, we investigate a range of methods—both prompt-based and fine-tuned—for enhancing LLM robustness to perturbed instructions in classification tasks. We focus on word- and character-level perturbations, as these have been found to cause the greatest performance decline (Zhu et al., 2024). We assess each method on a combination of six instructions, two types of perturbations, three datasets, and two base models. Our experiments show that LLMs are particularly effective at self-denoising instructions, especially when the process is done iteratively.

Our main findings are as follows: (1) iterative self-denoising—whether carried out by a fine-tuned model or the base LLM—prevents a considerable portion of the performance drop caused by using

*Equal contribution. Correspondence to contact@aryanagrawal.com, lisa.alazraki20@imperial.ac.uk.



(a) Non-perturbed. (b) DeepWordBug. (c) TextFooler.

Figure 1: Example perturbations of an instruction for sentiment classification, shown in (a). The perturbation can be at the character level, as shown in (b), or at the word level, as shown in (c).

perturbed instructions in classification tasks; (2) self-denoising in general is far more effective than other methods including instruction ensembling and hidden representation alignment; (3) other denoising strategies, such as perplexity smoothing, are not as successful. In fact, they tend to decrease performance further.

2 METHODS

We compare several methods to enhance LLM robustness to instruction perturbations. Namely, (iterative) self-denoising, perplexity smoothing, instruction ensembling, representation alignment. We have chosen these methods because they represent intrinsically different techniques to improve robustness. Further implementation details and hyperparameters can be found in Appendix A.

2.1 SELF-DENOISING

In self-denoising, we ask the LLM to unperturb a given instruction, given a meta-prompt and a set of in-context learning (ICL) examples (shown in Appendix B). Throughout the paper, we abbreviate this method as SD. We additionally explore variants of this method, described below.

Iterative self-denoising (SDi) This variant progressively unperturbs an instruction over multiple calls to the LLM, using the same meta-prompt and examples. If the instruction has not changed from the previous iteration, the process is stopped. Otherwise, the process will stop after five iterations.

Supervised fine-tuned self-denoising (SFT-SD) We fine-tune an LLM for the task of unperturbing instructions. We perform parameter-efficient tuning by adding LoRA modules (Hu et al., 2022) to the value and query projection layers of the frozen LLM. We create a novel training dataset, AdvMix, containing 2,900 pairs of perturbed and unperturbed sequences extracted from AdvGLUE (Wang et al., 2022b) and PromptBench (Zhu et al., 2024). Details of AdvMix are given in Appendix C. During both training and inference, the model observes the self-denoising meta-prompt and ICL examples. Note that the fine-tuned model can be applied in an iterative fashion at test time (SFT-SDi).

2.2 PERPLEXITY SMOOTHING

Inspired by randomised smoothing methods (Cohen et al., 2019; Gietz & Kalita, 2024; Robey et al., 2024; Zhang et al., 2024), we build a framework that minimises perplexity (PPL) as a proxy metric for the integrity of an instruction. Firstly, we rank words within an instruction by importance (leaving out stop words and class labels), where the importance of a word w_j is a function of the change in PPL of the instruction when w_j is deleted. We then mask the n top-ranked w_j with a `[MASK]` token, and adopt a masked language model to generate k candidate words to fill the mask. Having generated k variants of the instruction containing each candidate word, we select the β lowest-PPL variants and repeat the procedure masking the next-ranked word. We take the final, PPL-smoothed instruction resulting from this beam search process as the denoised instruction.

2.3 INSTRUCTION ENSEMBLING

We ensemble n variations of an instruction, each obtained by sampling with temperature from an LLM, using the same meta-prompt and ICL examples as in the self-denoising pipeline. We run

Algorithm 1: Greedy Search for Optimal Perturbation

Input: input instruction i, continuous goal function $\mathcal{G}$, set of transformations $\mathcal{T}$, set of constraints $\mathcal{C}$, query limit q_{max}
Output: optimal perturbed instruction i^*

$i^* \leftarrow i$
$q \leftarrow 0$
while $q < q_{max}$ **do**
  $\mathcal{I} \leftarrow \{\ \}$
  for $T \in \mathcal{T}$ **do**
    $i' \leftarrow T(i^*)$
    if $C(i')$ *is satisfied* $\forall C \in \mathcal{C}$ **then**
      $\mathcal{I} \leftarrow \mathcal{I} \cup \{i'\}$
  if $\mathcal{I} = \emptyset$ **then**
    `// No valid transformations from current` i^*
    break
  $n \leftarrow \min(q_{max} - q, |\mathcal{I}|)$
  $i^* \leftarrow \arg\max_{i' \in \mathcal{I}_{1:n}} \mathcal{G}(i')$
  $q \leftarrow q + |\mathcal{I}_{1:n}|$
return i^*

inference on each data sample using all n variations, and select the final classification label by majority vote.

2.4 Representation Alignment

For comparison, we implement a framework to align the hidden representation of the perturbed instruction to that of the non-perturbed one, similar to Sun et al. (2024). Given a dataset $\mathcal{D} = \{(i_j, i'_j)_{1 \leq j \leq N}\}$ containing pairs of unperturbed and perturbed instructions, we add LoRA adapter modules to a frozen LLM and train with the objective to minimise the cosine distance between $h(i_j)$ and $h(i'_j)$, where $h(i)$ is the hidden representation of i at the middle layer of the LLM. We use AdvMix for training. We choose the middle layer as a trade-off between capturing basic semantics (potentially useful for simpler perturbations, such as character-level edits) and representing contextual meaning (which may be relevant for more complex, word-level perturbations).

3 Experiments

We run all experiments with two well-known open-weight LLMs—Llama 3 8B Instruct (Dubey et al., 2024) and Flan-T5 Large (Chung et al., 2024). We refer to these models as Llama 3 and Flan-T5.

3.1 Perturbations

Given an instruction i, we obtain its perturbed version using a framework adapted from Morris et al. (2020)'s TextAttack. We greedily search for an optimal perturbation among the space of all possible perturbations $\mathcal{I}$, given a goal function $\mathcal{G}$. The search strategy is illustrated in Algorithm 1. Note that unlike in Morris et al. (2020), we do not implement early stopping upon $\mathcal{G}(i')$ reaching a threshold. The optimal perturbation can thus be defined as

$$i^* = \arg\max_{i' \in \mathcal{I}} C(i')\mathcal{G}(i'),$$

where $C(\cdot)$ is an indicator function, returning 1 when adhering to the constraints. In our case, the goal of the attack is to maximise the performance drop produced by the perturbed instruction. The

Table 1: Performance Drop Rate (PDR) obtained with perturbed instructions, aggregated by perturbation type, model and dataset. Lower PDR scores are better. For each method, we also report the average PDR improvement, i.e., the overall percentage change in PDR from the base LLM.

	PDR (↓)							**Avg. PDR improvement (↑)**
	Perturbation		**Model**		**Dataset**			
	TextFooler	DeepWord-Bug	Llama 3	Flan-T5	CoLA	QNLI	SST-2	
Base LLM	0.174	0.077	0.192	0.059	0.102	0.140	0.134	–
PPL smoothing	0.182	0.110	0.214	0.078	0.115	0.150	0.172	−16.3%
Instr. ensembling	0.130	0.037	0.142	0.026	0.071	0.094	0.086	33.3%
Repr. alignment	0.113	0.053	0.125	0.041	**0.052**	0.117	0.080	33.8%
SD	0.130	0.016	0.122	0.025	0.057	0.085	0.077	41.7%
SDi	0.125	**0.015**	0.119	0.021	0.053	0.091	0.065	44.3%
SFT-SDi	**0.072**	0.030	**0.082**	**0.021**	0.055	**0.062**	**0.036**	**59.2%**

constraints are that stop words and class labels must remain unperturbed. We choose perturbations that have been found to cause substantial performance degradation in previous literature (Zhu et al., 2024). These include character-level substitutions, insertions and deletions (Figure 1b) obtained with DeepWordBug (Gao et al., 2018), and word replacements by counter-fitted GloVe embeddings (Pennington et al., 2014) (Figure 1c) obtained using TextFooler (Jin et al., 2020).

3.2 Datasets

We evaluate the LLMs on three classification tasks from the GLUE benchmark (Wang et al., 2018). On these, base models achieve strong results, yet perturbing the instruction causes substantial performance loss (Zhu et al., 2024). The tasks are: (1) CoLA (Warstadt et al., 2019), which consists of 1k texts labelled as 'acceptable' or 'unacceptable' from a grammatical standpoint, (2) QNLI (Rajpurkar et al., 2016), a natural language inference dataset containing 5.5k samples, (3) SST-2 (Socher et al., 2013), comprising 1.8k text samples for binary sentiment analysis extracted from movie reviews. For each test set, we use six zero-shot instructions from the PromptBench library (Zhu et al., 2024), split among task-oriented and role-oriented. Instructions are shown in Appendix D.

3.3 Metric

We evaluate the efficacy of the methods with Performance Drop Rate (PDR) (Zhu et al., 2024). This metric measures the degradation in performance (i.e., classification accuracy) of an LLM under a perturbation, hence lower PDR values are better. Given a perturbation P, an instruction i, a robustness augmentation Φ, a base model f_θ, and a dataset of samples $\mathcal{D}_s = \{(x_j, y_j)_{1 \leq j \leq N}\}$, we compute the PDR as

$$\text{PDR}(P, i, \Phi, f_\theta, \mathcal{D}_s) = 1 - \frac{\sum_{j=1}^{N} \mathbb{1}\{\Phi(f_\theta, P(i), x_j) = y_j\}}{\sum_{j=1}^{N} \mathbb{1}\{\Phi(f_\theta, i, x_j) = y_j\}}.$$

Note that the performance discrepancy between the base LLMs and their Φ augmented versions is negligible when the instruction is clean (see Table 3 in Appendix E). This holds true for all the methods in Section 2, as none of them make substantial changes to a non-perturbed instruction.

3.4 Results and Analysis

Table 1 displays the PDR scores for each method, aggregated by perturbation type, model and dataset. We find that SFT-SDi is the best performing strategy overall, with 59.2% average PDR

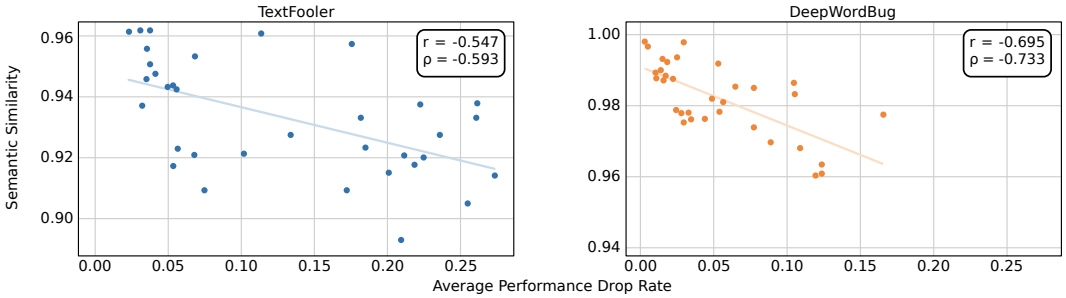


Figure 2: PDR and semantic similarity for TextFooler and DeepWordBug, averaged across models, datasets and instruction variants. For semantic similarity, we use the cosine similarity between the 4096-dimensional sentence embeddings encoded by E5 Mistral (Wang et al., 2024c). We choose this model since, at the time of writing, it achieves leading performance on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023), which is designed to evaluate the quality of text embeddings on a variety of tasks, including semantic similarity and text classification.

improvement. Generally, self-denoising achieves low PDR scores, even in the non-fine-tuned iterative setting (SDi, obtaining 44.3% avg. PDR improvement) and the vanilla SD setting (41.7%). We observe that instruction ensembling and representation alignment perform fairly similarly overall (33.3% and 33.8% avg. PDR improvement, respectively). We also find that PPL smoothing results in a performance decrease. This method increases the PDR over the base LLM in all cases, which reflects a negative PDR improvement.

It is worth noting that the lower PDR improvement given by representation alignment and instruction ensembling is mostly due to their PDR scores on DeepWordBug perturbations (.053 and .037 respectively, vs .016 for SD). On TextFooler, on the other hand, representation alignment achieves better PDR than both SD and SDi (though not SFT-SDi), while ensembling obtains a comparable PDR. In Figure 2, we analyse the effects of both perturbation types. We observe that TextFooler produces instructions that are semantically less similar to the original compared to DeepWordBug. This suggests that representation alignment and ensembling are effective when perturbed instructions substantially diverge semantically from the original, but they may be unsuitable for more subtle perturbations. Finally, we observe that semantic similarity is negatively correlated with PDR, with a stronger negative correlation for DeepWordBug perturbations ($r = -0.695$, $\rho = -0.733$) compared to TextFooler ($r = -0.547$, $\rho = -0.593$), suggesting that greater semantic deviation between the perturbed and original instructions leads to higher performance degradation.

4 Conclusion

We have investigated an extensive range of methods to enhance LLM robustness to perturbed instructions, across multiple models, datasets, perturbations and instruction templates. We have found that self-denoising—even in its simplest form—performs better on average than other methods. This highlights the ability of LLMs to self-correct perturbations to their instructions. We also observed that perplexity smoothing is completely ineffective at reducing PDR, causing instead a further loss in performance. Our empirical study lays substantial groundwork for the underexplored domain of LLM robustness to instruction perturbations, highlighting the most promising methods. Future research can further build upon these strategies, potentially investigating tasks beyond classification, larger model sizes and more complex perturbations such as semantic paraphrasing.

Acknowledgments

We would like to thank Thomas Mensink for the valuable advice he offered throughout this work, from its inception to the write-up. We also thank Fantine Huot, who provided many insightful comments on the first draft of this paper.

REFERENCES

Lisa Alazraki, Lluis Castrejon, Mostafa Dehghani, Fantine Huot, Jasper Uijlings, and Thomas Mensink. How (not) to ensemble LVLMs for VQA. In Javier Antorán, Arno Blaas, Kelly Buchanan, Fan Feng, Vincent Fortuin, Sahra Ghalebikesabi, Andreas Kriegler, Ian Mason, David Rohde, Francisco J. R. Ruiz, Tobias Uelwer, Yubin Xie, and Rui Yang (eds.), *Proceedings on "I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models" at NeurIPS 2023 Workshops*, volume 239 of *Proceedings of Machine Learning Research*, pp. 1–20. PMLR, 16 Dec 2023. URL https://proceedings.mlr.press/v239/alazraki23a.html.

Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, David Wagner, and Chuan Guo. SecAlign: Defending against prompt injection with preference optimization, 2025. URL https://arxiv.org/abs/2410.05451.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cohen19c.html.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pp. 177–190, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3540334270. doi: 10.1007/11736790_9. URL https://doi.org/10.1007/11736790_9.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLORA: efficient finetuning of quantized LLMs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri,

Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan

Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56, 2018. doi: 10.1109/SPW.2018.00016. URL https://ieeexplore.ieee.org/document/8424632.

Harrison Gietz and Jugal Kalita. MaskPure: Improving defense against text adversaries with stochastic purification. In *Natural Language Processing and Information Systems: 29th International Conference on Applications of Natural Language to Information Systems, NLDB 2024, Turin, Italy, June 25–27, 2024, Proceedings, Part I*, pp. 379–393, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-70238-9. doi: 10.1007/978-3-031-70239-6_26. URL https://doi.org/10.1007/978-3-031-70239-6_26.

Jiasheng Gu, Hongyu Zhao, Hanzi Xu, Liangyu Nie, Hongyuan Mei, and Wenpeng Yin. Robustness of learning from task instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13935–13948, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.875. URL https://aclanthology.org/2023.findings-acl.875/.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. Putnam-AXIOM: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=YXnwlZe0yf.

Shahin Honarvar, Mark van der Wilk, and Alastair Donaldson. Turbulence: Systematically and automatically testing instruction-tuned large language models for code, 2025. URL https://arxiv.org/abs/2312.14856.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-Young Paik, and Liming Zhu. Prompt perturbation in retrieval-augmented generation based large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 1119–1130, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671932. URL https://doi.org/10.1145/3637528.3671932.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL https://aclanthology.org/N18-1170/.

D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI); The Thirty-Second Innovative Applications of Artificial Intelligence Conference (IAAI); The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI)*, pp. 8018–8025. AAAI Press, February 2020. doi: 10.1609/aaai.v34i05.6311. URL https://aaai.org/ojs/index.php/AAAI/article/view/6311.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. TextBugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*, NDSS 2019. Internet Society, 2019. doi: 10.14722/ndss.2019.23138. URL http://dx.doi.org/10.14722/ndss.2019.23138.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6193–6202, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.500. URL https://aclanthology.org/2020.emnlp-main.500/.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681. URL https://aclanthology.org/2024.tacl-1.52/.

Milad Moradi and Matthias Samwald. Evaluating the robustness of neural language models to input perturbations. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1558–1570, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.117. URL https://aclanthology.org/2021.emnlp-main.117/.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.16. URL https://aclanthology.org/2020.emnlp-demos.16/.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle (eds.), *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1198/.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of GPT-4 on medical challenge problems, 2023. URL https://arxiv.org/abs/2303.13375.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: Defending large language models against jailbreaking attacks, 2024. URL https://arxiv.org/abs/2310.03684.

Melissa Roemmele and Andrew Gordon. From test-taking to test-making: Examining LLM authoring of commonsense assessment items. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5193–5203, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.299. URL https://aclanthology.org/2024.findings-emnlp.299.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. Natural language understanding with the Quora Question Pairs dataset, 2019. URL https://arxiv.org/abs/1907.01041.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Jiuding Sun, Chantal Shaib, and Byron C Wallace. Evaluating the zero-shot robustness of instruction-tuned language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g9diuvxN6D.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Evaluating adversarial attacks against multiple fact verification systems. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2944–2953, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1292. URL https://aclanthology.org/D19-1292/.

Candace Walkington, Virginia Clinton-Lisell, and Anthony Sparks. The effect of language modification of mathematics story problems on problem-solving in online homework. *Instructional Science*, 47, 10 2019. doi: 10.1007/s11251-019-09481-6. URL https://www.jstor.org/stable/48699919.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Bin Wang, Chengwei Wei, Zhengyuan Liu, Geyu Lin, and Nancy F. Chen. Resilience of large language models for noisy instructions. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11939–11950, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.697. URL https://aclanthology.org/2024.findings-emnlp.697/.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6134–6150, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.495. URL https://aclanthology.org/2020.emnlp-main.495/.

Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. SemAttack: Natural textual attacks via different semantic spaces. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 176–205, Seattle, United States, July 2022a. Association for Computational

Linguistics. doi: 10.18653/v1/2022.findings-naacl.14. URL https://aclanthology.org/2022.findings-naacl.14/.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models, 2022b. URL https://arxiv.org/abs/2111.02840.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xing Xie. On the robustness of ChatGPT: An adversarial and out-of-distribution perspective. *IEEE Data Eng. Bull.*, 47(1):48–62, 2024b. URL http://sites.computer.org/debull/A24mar/p48.pdf.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models, 2024c. URL https://arxiv.org/abs/2401.00368.

Xuezhi Wang, Haohan Wang, and Diyi Yang. Measure and improve robustness in NLP models: A survey. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4569–4586, Seattle, United States, July 2022c. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.339. URL https://aclanthology.org/2022.naacl-main.339/.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040/.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Lingling Xu, Haoran Xie, Zongxi Li, Fu Lee Wang, Weiming Wang, and Qing Li. Contrastive learning models for sentence representations. *ACM Trans. Intell. Syst. Technol.*, 14(4), June 2023. ISSN 2157-6904. doi: 10.1145/3593590. URL https://doi.org/10.1145/3593590.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6066–6080, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.540. URL https://aclanthology.org/2020.acl-main.540/.

Xinyu Zhang, Hanbin Hong, Yuan Hong, Peng Huang, Binghui Wang, Zhongjie Ba, and Kui Ren. Text-CRS: A Generalized Certified Robustness Framework against Textual Adversarial Attacks . In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2920–2938, Los Alamitos, CA, USA, May 2024. IEEE Computer Society. doi: 10.1109/SP54263.2024.00053. URL https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00053.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. PromptBench: A unified library for evaluation of large language models. *J. Mach. Learn. Res.*, 25:254:1–254:22, 2024. URL https://jmlr.org/papers/v25/24-0023.html.

A Implementation Details

A.1 Self-denoising

For vanilla self-denoising (SD) and iterative self-denoising (SDi), we use greedy decoding with temperature $t = 0$.

In Table 2 we detail the hyperparameters used for training the LoRA modules in the fine-tuned self-denoising pipeline (SFT-SD) on AdvMix. The same hyperparameters are used for both Llama 3 and Flan-T5. Note that we use the hyperparameter combination from Dettmers et al. (2023), since this has been shown to generalise well to a wide range of tasks.

A.2 Perplexity Smoothing

All PPL scores in perplexity smoothing are computed using GPT-2 (Radford et al., 2019). Candidate substitute words for each `[MASK]` token are found using DistilRoberta Base[1]. In our experiments, we set $n = 10$ (i.e., we mask the top ten most important words). We also set the same beam width β as the number of candidates k, i.e. $k = \beta = 5$, thus performing best-first search.

A.3 Instruction Ensembling

For instruction ensembling, we sample n options from the LLM with temperature $t = 1$, and set $n = 5$. As the classification tasks in our experimental setup are binary, this value of n ensures that it is always possible to take the majority label as the final classification label.

A.4 Representation Alignment

We use a siamese model implementation (Chen et al., 2020; Xu et al., 2023; Sun et al., 2024) to align the hidden representations of the perturbed instructions to those of the non-perturbed instructions. We align representations at the layer l, where l is chosen to be the middle layer of the LLM. For Llama 3 8B Instruct (32 decoder layers), we set $l = 16$. For Flan-T5, we take advantage of the encoder-decoder architecture and set l to be last hidden layer of the encoder block. The siamese network is trained on the instruction pairs in AdvMix using LoRA modules (Hu et al., 2022) at the value and query projection layers of the LLM. The LoRA modules are disabled or enabled during each forward pass depending on whether the input consists of unperturbed or perturbed instructions, respectively. Since unperturbed and perturbed instructions may differ in token count, mean pooling is applied to their middle-layer hidden representations before computing the cosine distance loss.

The training hyperparameters—for both the Llama 3 and the Flan-T5 implementation—are shown in Table 2 (note that the same hyperparameters are used for training the SFT self-denoising models).

[1] https://huggingface.co/distilbert/distilroberta-base

Table 2: Hyperparameters for training the SFT self-denoising models and the representation alignment network. The hyperparameter combination is the same for both base LLMs (Llama 3 and Flan-T5).

Hyperparameter	**Value**
LoRA α	16
LoRA r	64
LoRA dropout	0.1
LoRA modules	$Q_{\text{proj}}, V_{\text{proj}}$
Learning rate	5e−5
Batch size (effective)	4
Epochs	10

B SELF-DENOISING META-PROMPT AND EXAMPLES

In Prompt 1, we show the meta-prompt and few-shot examples used at both training and inference in the self-denoising pipeline. All examples are extracted from MNLI (Williams et al., 2018), as these achieved the highest validation results across the different datasets, surpassing setups where the exemplar instructions were extracted from multiple diverse tasks.

Prompt 1: Meta-prompt and examples for self-denoising

```
Given a sentence which could be perturbed through an adversarial
attack, respond with the unperturbed sentence.  Do not modify the
following words:  {excluded_words}.  Do not answer with anything
other than the unperturbed sentence.



Uncovering whether the made coupling of condemns revealed
entailment, neutral, or contradiction.  Cope with 'entailment',
'neutral', or 'contradiction':

Identify whether the given pair of sentences demonstrates
entailment, neutral, or contradiction.  Answer with 'entailment',
'neutral', or 'contradiction':


Specifies if the made coupling of condemns exposure entailment,
neutral, or contradiction.  Reacting with 'entailment', 'neutral',
or 'contradiction':

Determine if the given pair of sentences displays entailment,
neutral, or contradiction.  Respond with 'entailment', 'neutral',
or 'contradiction':


Can the ratio between the offered penalty be entailment, neutral,
or contradiction?  Reactions with 'entailment', 'neutral', or
'contradiction':

Does the relationship between the given sentences represent
entailment, neutral, or contradiction?  Respond with 'entailment',
'neutral', or 'contradiction':
```

C TRAINING DATA

We train the SFT-SD model and the representation alignment pipeline on AdvMix, a custom dataset containing 2,882 pairs of unperturbed and perturbed text sequences.

To create AdvMix, we extract 2,530 pairs of sequences (88% of the total) from the AdvGLUE benchmark (Wang et al., 2022b), which contains the original and perturbed versions of the samples in GLUE (Wang et al., 2018). Our AdvGLUE selection is split among Quora Question Pairs (QQP) (Sharma et al., 2019), Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018), and Recognizing Textual Entailment (RTE) (Dagan et al., 2005). Note that none of these datasets are used at test time.

Since AdvGLUE contains general text sequences rather than instructions, we augment AdvMix with 352 samples from PromptBench (Zhu et al., 2024), which contains perturbed and unpreturbed instruction prompts. Again, we are careful to use use datasets that do not appear among our testing data. We extract from Microsoft Research Paraphrase Corpus (MRPC) (Dolan & Brockett, 2005).

Note that both AdvGLUE and PromptBench include more than one perturbation for each clean text sequence. To aid the generalization of our train model, we include these different perturbations in AdvMix. We thus source sequences perturbed via four word-level attacks (BERT-Attack (Li et al., 2020), SememePSO (Zang et al., 2020), TextBugger (Li et al., 2019), TextFooler (Jin et al., 2020)) and seven sentence-level attacks (CheckList (Ribeiro et al., 2020), AdvFever (Thorne et al., 2019), SCPN (Iyyer et al., 2018), SemAttack (Wang et al., 2022a), T3 (Wang et al., 2020), StressTest (Naik et al., 2018), and DeepWordBug (Gao et al., 2018)).

D INSTRUCTIONS

We use six instruction variants for each dataset, equally split among *task-oriented* and *role-oriented* instructions.

D.1 COLA INSTRUCTIONS

Task-Oriented

```
Assess the following sentence and determine if it is grammatically
correct.  Respond with 'acceptable' or 'unacceptable':
```

```
Examine the given sentence and decide if it is grammatically sound.
Answer with either 'acceptable' or 'unacceptable':
```

```
Analyze the provided sentence and classify its grammatical correctness as
'acceptable' or 'unacceptable':
```

Role-Oriented

```
In your role as a grammar check tool, assess the following sentence
and classify it as 'acceptable' if it is grammatically correct or
'unacceptable' if it is incorrect:
```

```
As a grammar identification system, examine the provided sentence
and respond with 'acceptable' for grammatically correct sentences or
'unacceptable' for incorrect ones:
```

```
Functioning as a grammar evaluation tool, analyze the given sentence and
decide if it is grammatically correct, responding with 'acceptable' or
'unacceptable':
```

D.2 QNLI INSTRUCTIONS

Task-Oriented

Given the question and context provided, determine if the answer can be inferred by choosing 'entailment' or 'not_entailment'.

Based on the provided context and question, decide if the information supports the answer by responding with 'entailment' or 'not_entailment'.

Please assess if the answer to the question can be derived from the given context by selecting 'entailment' or 'not_entailment'.

Role-Oriented

As a language expert, assess if the given context entails the answer to the question and respond with 'entailment' or 'not_entailment'.

In your role as a semantic evaluator, determine if the provided context justifies the answer to the question and answer with 'entailment' or 'not_entailment'.

As a textual analyst, examine if the given context logically implies the answer to the question and indicate your decision with 'entailment' or 'not_entailment'.

D.3 SST-2 INSTRUCTIONS

Task-Oriented

Analyze the tone of this statement and respond with either 'positive' or 'negative':

Evaluate the sentiment of the given text and classify it as 'positive' or 'negative':

Please identify the emotional tone of this passage: 'positive' or 'negative'?

Role-Oriented

As a sentiment classifier, determine whether the following text is 'positive' or 'negative'. Please classify:

In the role of a sentiment analysis tool, respond with 'positive' or 'negative' to classify this statement:

Acting as a sentiment evaluator, identify if the given sentence is 'positive' or 'negative'. Classify:

E PERFORMANCE WITH NON-PERTURBED INSTRUCTIONS

In Table 3 we show the classification accuracy on non-perturbed instructions for each method. Scores are averaged across datasets (CoLA, QNLI, SST-2), underlying LLMs (Llama 3, Flan-T5) and instruction variants (six variants for each dataset). Note that the accuracy scores obtained by the augmented pipelines are within only 1% of that achieved using the base model implementation.

Table 3: Accuracy scores for each method, averaged across datasets and LLMs.

Method	Avg. performance
Base LLM	80.1
PPL smoothing	79.3
Instruction ensembling	80.0
Representation alignment	79.8
SD	80.0
SDi	80.0
SFT-SDi	79.1