

UCLA

UCLA Electronic Theses and Dissertations

Title

From Metrics to Meaning: Advancing Evaluation Frameworks for Robust and Human-Centric AI

Permalink

<https://escholarship.org/uc/item/8qs5j32q>

Author

Harel-Canada, Fabrice

Publication Date

2025

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

From Metrics to Meaning: Advancing Evaluation Frameworks for Robust and Human-Centric AI

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Fabrice Harel-Canada

2025

© Copyright by
Fabrice Harel-Canada
2025

ABSTRACT OF THE DISSERTATION

From Metrics to Meaning: Advancing Evaluation Frameworks for Robust and Human-Centric AI

by

Fabrice Harel-Canada

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2025

Professor Miryung Kim, Chair

The rapid evolution of machine learning (ML) presents significant evaluation challenges, particularly regarding data quality (data bugs) and the limited applicability of traditional software testing metrics. Standard evaluation often overlooks crucial aspects like nuanced data diversity, subjective user experience, and content provenance. This dissertation argues for a paradigm shift in ML evaluation, moving beyond narrow, structural metrics towards holistic frameworks. The central thesis is that robust and meaningful assessment requires integrating complementary perspectives: systematically enhancing data diversity, incorporating human-centric subjective evaluation, and ensuring reliable authorship verification.

This work investigates these pillars through empirical studies. Initial analysis reveals the limitations of structural metrics like neuron coverage (NC) for effectively guiding diverse test generation in deep neural networks (RQ1). Subsequently, we demonstrate that aggressively expanding input diversity via label-altering transformations (sibylvariance) significantly improves model generalization, defect detection, and robustness (RQ2). To refine augmentation strategies, we introduce feature-aware data augmentation (FADA) to optimize policies by balancing diversity and text quality, alongside INSPECTOR, a human-in-the-loop system using transformation provenance for efficient data validation (RQ3). Shifting focus to output quality, we propose the Psychological Depth Scale (PDS) to measure subjective, human-centric dimensions like empathy and engagement, find-

ing that current language models can achieve high, human-like levels (RQ4). Recognizing the challenge this poses, we analyze watermarking techniques, demonstrating their practical resilience against attacks and affirming their necessity for authorship verification in the age of advanced AI (RQ5).

Collectively, these findings underscore the need for a multi-faceted evaluation approach. This research contributes components towards a unified framework synthesizing structural rigor, data-centric diversity, human judgment on quality and subjective experience, and technical methods for provenance. By adopting such holistic evaluation drawing on software engineering, machine learning, and human-computer interaction we can better assess ML systems for real-world reliability, utility, and trustworthiness, fostering the development of more beneficial and accountable AI.

The dissertation of Fabrice Harel-Canada is approved.

Quanquan Gu

Muhammad Ali Gulzar

Amit Sahai

Nanyun (Violet) Peng

Miryung Kim, Committee Chair

University of California, Los Angeles

2025

To future researchers and to all they will sacrifice to do good work

TABLE OF CONTENTS

1	Introduction	1
1.1	Research Problem	1
1.2	Is Neuron Coverage a Meaningful Metric for Testing Deep Neural Networks? . . .	3
1.3	Data Augmentation and its Impact on Data Diversity and Model Performance . . .	4
1.3.1	Sibylvariant Transformations for Robust Text Classification	4
1.3.2	FADA: Learning Effective Data Augmentation Policies with Text Quality Metrics and Semantic Features	6
1.3.3	Human-in-the-Loop Synthetic Text Data Inspection with Provenance Tracking (INSPECTOR)	7
1.4	Measuring Psychological Depth in Language Models (PDS)	8
1.5	Sandcastles in the Storm: Revisiting the (Im)Possibility of Strong Watermarking .	9
1.6	A Unified Framework for Evaluation	10
2	Background	13
2.1	Testing and Coverage Metrics for Deep Learning	13
2.2	Data Augmentation and Transformation Paradigms	14
2.3	Model Debugging and Synthetic Data Inspection	16
2.4	Creativity Assessment in Generative AI	17
2.5	Watermarking and Content Provenance	18
3	Is Neuron Coverage a Meaningful Metric for Testing Deep Learning Networks? . .	20
3.1	Introduction	20
3.2	Related Work	23
3.3	Study Methods	26

3.3.1	Datasets and DNNs	26
3.3.2	Measuring Neuron Coverage	27
3.3.3	Adversarial Attack Algorithms	28
3.3.4	Extending Attacks to Increase NC	29
3.4	Findings	33
3.4.1	Defect Detection	34
3.4.2	Naturalness	34
3.4.3	Output Impartiality	38
3.5	Discussion	43
3.5.1	DeepXplore & DeepTest Comparison	44
3.5.2	How Meaningful Is a Neuron?	44
3.5.3	Does NC Maximization Make Sense for Testing DNNs?	45
3.6	Conclusion	46
4	Sibylvariant Transformations for Robust Text Classification	48
4.1	Introduction	48
4.2	Sibylvariance	50
4.2.1	Sibylvariant Subtypes	51
4.2.2	Adaptive Sibylvariant Training	52
4.3	Transformations	52
4.4	Experiments	54
4.4.1	Transformation Pipelines & Datasets	54
4.4.2	Model Setting	55
4.4.3	RQ1. Generalization Performance	56
4.4.4	RQ2. Defect Detection	58

4.4.5	RQ3. Adversarial Robustness	60
4.5	Discussion	61
4.6	Related Work	64
4.7	Conclusion	65
4.8	Appendix	66
4.8.1	Implemented Sybil Transformations	66
4.8.2	Other Possible Text Transformations	68
4.8.3	Sibylvariant Subtype Examples	71
4.8.4	Detailed Training Results for RQ1	72
4.8.5	Detailed Defect Detection Results for RQ2	74
4.8.6	Detailed Robustness Results for RQ3	75
5	FADA: Learning Effective Data Augmentation Policies with Text Quality Metrics and Semantic Features	76
5.1	Introduction	77
5.2	Motivating Text Quality as Predictive of Augmentation Effectiveness	79
5.3	Approach	80
5.3.1	FADA	80
5.3.2	Text Transforms	83
5.3.3	Text Features	83
5.3.4	Quality Metrics	84
5.4	Experimental Setup	85
5.4.1	Benchmark Datasets	85
5.4.2	Baselines	85
5.4.3	Text Classification Models	86

5.5	Experimental Results	87
5.5.1	Augmentation’s Impact on Text Quality	87
5.5.2	Generalization Performance	88
5.5.3	Correlations Between Text Quality and Post-Train Generalization	88
5.5.4	Transferability of FADA Policies	90
5.6	Related Work	90
5.7	Conclusion	91
5.8	Appendix	92
5.8.1	Benchmark Dataset Statistics	92
5.8.2	Studied Transform Descriptions	92
5.8.3	Model and Training Settings	94
6	Human-in-the-Loop Synthetic Text Data Inspection with Provenance Tracking . .	95
6.1	Introduction	96
6.2	Design Goals and System Overview	98
6.2.1	Design Goals	98
6.2.2	Overview	99
6.2.3	Usage Scenario.	100
6.3	Implementation	102
6.3.1	Overview	102
6.3.2	Provenance Tracking	102
6.3.3	Assistive labeling	103
6.3.4	Study Design	105
6.4	Results	106
6.4.1	Reduction in Human Effort	106

6.4.2	User Ratings of Individual Features	107
6.4.3	Robustness	109
6.5	Implications	109
6.6	Related Work	110
6.6.1	Debugging Machine Learning	110
6.6.2	Cleaning Data	111
6.7	Conclusion	111
7	Measuring Psychological Depth in Language Models	113
7.1	Introduction	114
7.2	The Psychological Depth Scale	116
7.3	PSYCHDEPTH Dataset	119
7.3.1	Human Stories	120
7.3.2	LLM Stories	120
7.4	Human Study	121
7.5	Results	122
7.5.1	RQ1. Consistency of Human Judgments	122
7.5.2	RQ2. LLM-as-Judge for Measuring Psychological Depth	123
7.5.3	RQ3. Comparing Psychological Depth in Human and LLM Stories	125
7.6	Discussion	126
7.7	Related Work	127
7.7.1	Evaluating Creative Writing	127
7.7.2	Creative Generation by LLMs	128
7.8	Conclusion	129
7.9	Limitations and Risks	130

7.10	Appendix	132
7.10.1	Dataset Quality Control Measures	132
7.10.2	Prompt Premises	134
7.10.3	Story Examples	135
7.10.4	Authorship Reasons	148
7.10.5	Study Details	150
7.10.6	Visualizations of Depth Ratings	153
7.10.7	Mixture-of-Personas	156
7.10.8	Statistical Tests for Author Comparisons	156
8	Sandcastles in the Storm: Revisiting the (Im)possibility of Strong Watermarking	158
8.1	Introduction	159
8.2	Background	161
8.3	Evaluation Setup	163
8.4	Analysis	165
8.4.1	RQ1. Can Stationary Distributions Be Reached Under Practical Constraints?165	
8.4.2	RQ2. Are LLM-based quality oracles sophisticated enough to guide a random-walk attack?	166
8.4.3	RQ3. How effective are random walk attacks in breaking watermarks when controlling for quality?	169
8.5	Discussion	171
8.6	Conclusion	173
8.7	Appendix	173
8.7.1	Formal Definitions	173
8.7.2	Evaluation Setup	178
8.7.3	Watermark Details	178

8.7.4	Entropy-Controlled Prompt Dataset	181
8.7.5	Dataset Statistics	183
8.7.6	Perturbation Oracle Details	184
8.7.7	Distinguisher Prompt	185
8.7.8	A Challenging Distinguisher Example	187
8.7.9	Extended Distinguisher Study	193
8.7.10	Breakdown by Domain and Entropy	193
8.7.11	Quality Oracles	195
8.7.12	Full Oracle Results	197
8.7.13	InternLM vs DiffOracle	198
8.7.14	Human Annotation Details	198
8.7.15	Extended Attack Results Analysis	200
8.7.16	Attack Success Rate vs. Detection Threshold	202
8.7.17	What factors contributed to attack inefficiency?	203
9	Conclusion	210

LIST OF FIGURES

3.1	Single Layer DNN. ❶ represents inputs (i.e., pixels, features, etc.). ❷ represents a hidden layer of 5 neurons, where parentheses denote activations scaled between 0 and 1 for comparison against a NC threshold. ❸ represents an output layer of 1 neuron (i.e., class logits, probabilities, etc.).	28
3.2	Neural activation before and after regularization: our regularization significantly promotes NC at $t = 0.2$	29
3.3	$NC_{t=0.2}$ vs ASR: the results show that NC does not consistently correlate with defect detection.	32
3.4	$NC_{t=0.2}$ vs Naturalness (IS / FID): the results show both strongly negative and strongly positive correlations.	37
3.5	Test Suite #33. $NC_{t=0.2}$: 0.29 - IS: 1.97 - FID: 0.10	38
3.6	Test Suite #140. $NC_{t=0.2}$: 0.33 - IS: 1.48 - FID: 2.96	38
3.7	$NC_{t=0.2}$ vs Output Impartiality: the results show that increasing NC creates bias in output behavior.	39
3.8	Output Prediction Distribution Histogram (left) and Cross-Class Prediction Heatmap (right)	43
3.9	Visualization of neuron activations shows mixed concepts—cats, foxes, and cars [227]	45
4.1	C2S intakes a text and its label (red) to extract keywords, ['stupid, worse']. These words are used to generate a new INV sentence shown in red. Alternatively, antonym (left) and synonym (right) substitution can produce new concepts that further boost diversity.	54
4.2	SIB* outperforms INV* most, when data availability is low, indicating the necessity of SIB to complement INV.	56

4.3	The best performing TP for each dataset trained on 200 examples per class. α SentMix or α TextMix leads to the highest performance gains. SIB* consistently outperforms INV*.	57
4.4	RQ2 defect detection comparison. Percentages show change in accuracy relative to T_{ORIG} . Lower accuracy indicates greater efficacy at inducing error.	60
4.5	UMAP visualizations of BERT [CLS] tokens for SST-2. Blue, red, and green represent “Negative,” “Positive,” and “Mixed”, respectively.	62
5.1	Quality metric correlations with F1-scores of already fine-tuned, open-source models. .	80
5.2	Overview of the FADA search procedure. In this sentiment analysis example, applying transform t_i to a data subset D_{f_i} containing feature f_i , results in adverse impacts to average alignment, fluency, and grammaticality. The transform-feature interaction is summarized into the augmentation policy with a score of 0.1, which indicates a relatively low probability of sampling t_i for any texts containing f_i	81
5.3	Quality metric correlations between each other and test F1-scores of models trained with augmented data, but tested on unagumented data.	89
5.4	Transferability of FADA policies learned from a source dataset and applied to other target datasets. The numbers denote changes in F1-scores, averaged across both model architectures.	90
6.1	Examples of transformed texts from the SST2 movie review dataset generated during data augmentation. A transformed text can contain garbled text, or have an inappropriate label. As an example, the “Word Deletion” transformation can mangle the text “ends up being surprisingly dull” into “up being surprising”, causing its corresponding label “-” (indicating a negative sentiment) to no longer be appropriate. Of the four examples of synthetically generated texts, only one (“the event is beautiful to see”) has an appropriate label.	96

6.2	INSPECTOR: The user alternates between (1) inspecting the provenance of groups of texts and labels following their (A) common transformation, and (B) common linguistic features, and (2) inspecting individual transformed texts with their corresponding labels, with assistive labeling using (C) the quality metrics, alignment, grammaticality, fluency scores, and (D) LLM predictions.	97
6.3	Transform provenance. A user selects texts and inspects the common transforms (e.g., <code>RandomCharSubset</code>) in the transformation provenance pane with their (A) inspection statistics (e.g., the user has inspected 14 texts, with 11 marked as high quality), and (B) view other texts sharing the same transform. A user can then (C) mark all instances sharing the same transform to be correct, obviating the need for inspecting individual texts one by one.	99
6.4	Feature provenance. A user can select texts, and can inspect linguistic features common to the selected texts (e.g., “Has a description of a location”) in the transformation provenance pane with their inspection statistics (e.g., the user has inspected 24 texts, with 11 marked as high quality). Then, a user can mark all instances sharing the same feature to be correct.	100
6.5	The workflow of a user inspecting data using INSPECTOR.	102
6.6	INSPECTOR enables assistive labeling by providing (A) quality metrics such as label alignment, fluency, and grammaticality. Label alignment is a measure of label quality, while fluency and grammaticality are measures of linguistic quality. It also shows the (B) LLM predictions for a text compared to its corresponding label.	103
6.7	Most participants perceived INSPECTOR to be more helpful in winnowing out generated texts with incorrect labels (such as the examples in Figure 6.1).	106
6.8	Participants using INSPECTOR were more confident of their inspections, with a median rating of 5 compared to the baseline of 4.	106

7.1	In this GPT-4 story, the psychological depth scale highlights strengths and weaknesses contributing to the overall reader experience, providing additional quality signals over traditional metrics more likely to saturate. Scores are normalized 1-5 for comparison. .	114
7.2	Overview of our approach to developing and validating the Psychological Depth Scale. We merge related metrics from an extensive survey of literary theory and reader-response analysis, then generate deep stories using LLMs, and finally compare annotations from both human evaluators and automated systems across five key dimensions: authenticity, narrative complexity, empathy, engagement, and emotion provocation.	115
7.3	Illustration of WRITERPROFILE’s template, which prompts an LLM to generate stories based on a premise and a writer profile.	139
7.4	Illustration of PLAN+WRITE’s workflow, which prompts an LLM for character portraits given a premise prior to story generation.	142
7.5	Screenshots taken from the Warm-Up tutorial instructions shown to study participants. All fields are similar to the ones used in the main annotation forms.	151
7.6	Screenshots showing how annotations were collected for each story using Google Forms.	152
7.7	Cumulative Distribution Function (CDF) plots for each component of psychological depth. Steeper curves indicate a greater proportion of high ratings and overall stronger performance.	154
7.8	Spider plot comparing the psychological depth scores of 5 popular LLMs vs spectrum of human writers.	155
7.9	Heatmap comparing whether differences in author scores are statistically significant using pairwise t-tests. Color indicates the strength of the t-statistic, where higher (bluer) means the lefthand author scores are higher. Cell annotations represent p-values, where we regard $p < 0.05$ as statistically significant.	157
8.1	InternLM Quality Distribution by Watermarking Scheme W	179
8.2	Perplexity Distribution by Watermarking Scheme W	179
8.3	Unique Bigrams Distribution by Watermarking Scheme W	180

8.4	Grammar Errors Distribution by Watermarking Scheme W	180
8.5	Attack success rate (ASR) vs. detection threshold for the Adaptive watermarking scheme. Each curve represents a different perturbation oracle, with thresholds measured in standard deviations above the unwatermarked mean.	203
8.6	Attack success rate (ASR) vs. detection threshold for the SIR watermarking scheme. The plot shows the fraction of attacked texts falling below various thresholds (in standard deviations above the unwatermarked mean) for multiple perturbation oracles. . . .	204
8.7	Attack success rate (ASR) vs. detection threshold for the KGW watermarking scheme. Different curves correspond to various perturbation oracles, with the detection threshold defined as standard deviations above the mean detection score of unwatermarked texts.	205
8.8	Rolling success rate for GPT-4o generations, which are unwatermarked.	206
8.9	Rolling success rate for the KGW watermark.	207
8.10	Rolling success rate for the SIR watermark.	208
8.11	Rolling success rate for the Adaptive watermark.	209

LIST OF TABLES

3.1	DNN Architectural Details	27
3.2	Original NC, Average % Increase from Original NC, and Maximum % Increase from Original NC	30
3.3	Experimental Variables	31
3.4	Correlation between NC & ASR: Gray indicates a p-value > 0.05	33
3.5	Correlation between NC & Naturalness: Gray indicates a p-value > 0.05	36
3.6	Correlation between NC & Output Impartiality: Gray indicates a p-value > 0.05	40
3.7	MNIST Class and Average Rank of NC Baseline	42
4.1	Transformations currently available in SIBYL. New transforms that we defined are marked with $\dagger$	53
4.2	TP descriptions. TP s with an α -prefix use targeted, adaptive training (Section 4.2.2).	55
4.3	Dataset details. Test represents the number of examples per class in the test set.	55
4.4	RQ1 accuracy comparison for INV*, SIB*, and INVSIB against baselines ORIG, TMix [48], EDA [318], AEDA [135]. An asterisk (*) indicates the best performance observed across underlying TP s of each kind, while a $\ddagger$ indicates related works for comparison.	57
4.5	Top ten individual transforms over a no-transform baseline averaged across all datasets. The INV / SIB types were SIB for the sentiment analysis datasets and INV for the topic classification datasets. See Table 4.11 in the Appendix for more details.	58
4.6	RQ3 adversarial robustness comparison for INV*, SIB*, and INVSIB using TextFooler (TF), DeepWordBug (DWB), and TextBugger (TB). A lower attack success rate indicates a higher adversarial robustness.	61

4.7	Transform descriptions currently implemented in <code>Sybil</code> , sampled from according to task (sentiment analysis or topic) and <i>TP</i> (INV, SIB, or INVSIB). Note that transformations are INV or SIB with respect to specific tasks. Asterisks (*) indicate that the variance type could be either INV or SIB, but the listed variance was judged to be more likely.	67
4.8	Transform NOT currently implemented in <code>Sybil</code> , but represent potentially interesting directions for future work. Asterisks (*) indicate that the variance type could be either INV or SIB, but the listed variance was judged to be more likely.	69
4.9	Examples of SIB transformations for the image and text domains. For mixture mutations, we show a soft label proportional to the pixel and word counts of their constituent parts.	71
4.10	Performance (test set accuracy (%)) for all <i>TPs</i> . The results are averaged across three runs. Models are trained with either 10, 200, or 2500 examples per class. <i>TPs</i> are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively. White with a ‡ indicates related works for comparison. For TMix, EDA, and AEDA, we used the author’s open source code with their default / recommended configurations to transform the training datasets. However, we maintained the same model training hyperparameters as our other <i>TPs</i> to facilitate fair comparisons with our work.	72
4.11	Performance (test set accuracy (%)) for individual transforms over a no-transform baseline averaged across all datasets. The INV / SIB types were SIB for the sentiment analysis datasets and INV for the topic classification datasets.	73
4.12	Test suite accuracy (%) by dataset and <i>TP</i> . Lower accuracy indicates higher defect detection potential. <i>TPs</i> are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively.	74

4.13	Attack success by dataset and TP for three adversarial algorithms: TextFooler (TF), DeepWordBug (DWB), and TextBugger (TB). Lower attack success indicates higher adversarial robustness. TP s are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively.	75
5.1	Relative impact different augmentations have on our studied text quality metrics & Dist-2 as a comparative diversity metric, averaged across all datasets.	87
5.2	Test F1-score (%) with standard deviation of different augmentation approaches in a low-resource regime. Results are averaged across three runs. All approaches use a $3\times$ augmentation multiplier for fairer comparison.	88
5.3	Dataset statistics.	92
5.4	Descriptions of all the text transforms in the FADA search space. Sources: 1. SIBYL [109] 2. CHECKLIST [261] 3. TEXTATTACK [213] 4. AEDA [136] 5. EDA [316]. Note that source attributions are based on implementation details, not necessarily where the transformation was initially proposed.	93
6.1	The text transformations for generating data in the user study. For example, given the text “ends up being surprisingly dull” with a “negative” label, “Word Deletion” produces a new text “up being surprising” with the same corresponding label.	104
6.2	The two datasets in the user study. We analyze only the synthetic texts generated using the data augmentation techniques. During the course of the user study, 100% and 99% of the SST2 and TweetEval datasets were reviewed by at least one participant.	105
6.3	The average number of texts with correct labels identified. INSPECTOR improves over a baseline without provenance tracking and assistive labeling by $3\times$ and $4\times$. SA: Sentiment Analysis, HS: Hate Speech Detection	107
6.4	Usefulness of the techniques of INSPECTOR rated by the participants (out of 7). A higher rating indicates the technique was perceived to be more useful for inspection. Grouping texts by common transformations was perceived to be the most useful technique.	108

6.5	Results comparing the robustness of BERT after finetuning on additional data. We measure the attack success rate of DeepWord (the lower the better). SA: Sentiment Analysis, HS: Hate Speech Detection	109
7.1	Rater agreement on each PSD component as measured by Krippendorff’s alpha ($K-\alpha$). Mixture-of-Personas (MoP) prompting helps inject useful diversity of opinion into the annotations.	122
7.2	Zero-shot correlations between LLM-as-Judge and Humans on each PSD component where MoP indicates Mixture-of-Personas prompting. All correlations are significant at $p < 0.05$ except those with strikethrough	124
7.3	Average human ratings (5-point Likert) and standard deviations for each component of psychological depth, as well as HUM: the estimation of human or LLM authorship (1 is LLM and 5 is Human).	126
7.4	Average number of regeneration attempts before the story satisfied our length constraint of 400-600 words. WP stands for WRITERPROFILE prompting and P+W stands for PLAN+WRITE prompting.	132
7.5	Average probabilities of being AI generated as reported by Ghostbuster [305]. Results show that on average, human stories are unlikely to be authored by LLMs. Consistent with our authorship classification analysis, GPT-4 stories were also considered likely to be human-written.	134
7.6	All 15 premises we sourced from Reddits <code>r/WritingPrompts</code> to elicit psychological depth.	136
7.7	PSD Ratings for example stories show in the appendix. <code>p_id = premise_id</code> and <code>s_id = story_id</code>	138
7.8	Percent change in mean ratings when switching from WRITERPROFILE prompting to PLAN+WRITE prompting, aggregated by model and component. For example, PLAN+WRITE decreased Engagement for GPT-4 by 16% relative to using WRITERPROFILE.	148

7.9	Common reasons for LLM or human authorship decisions as a percent of stories receiving those comments.	149
7.10	Personas used with SYSTEM message tag to prime the LLM for a particular perspective relevant to annotation.	156
8.1	Summary of failed distinguisher tests per P , along with the step budget and total tests. Classification is first performed by Llama-3.1-70B, followed by GPT-4o on its failures, then o3-mini-high on any remaining cases. The overall 100% success rate indicates that the attacked texts never lose memory of their starting points, contradicting KA1 and suggesting that a stationary distribution is not reached in practice. . .	165
8.2	Performance of selected quality oracles on human-annotated data (full results in Appendix 8.7.11.1). QP Precision measures accuracy in preserving high-quality outputs, while Overall F1 reflects general classification performance. Despite fine-tuning, no oracle fully aligns with human judgments, challenging KA2 and limiting their reliability in guiding random-walk attacks.	167
8.3	Attack success rates (ASR) across different perturbation strategies. Human review reveals an average of 59.52% of successfully attacked texts have degraded quality. μ_{w_0} represents the initial watermark score at step 0, while μ_{w_t} represents the final watermark score after t mutation steps. “min” refers to the point where the watermark score is at its lowest during the attack while “fin” refers to score at the final step of the attack. “Reviewed” indicates the number of human-reviewed examples where the watermark was broken. QP and ¬QP represent the number of cases where human reviewers judged the attacked text as quality-preserving or degraded, respectively. Q-ASR_{fin} is the re-estimated attack success after controlling for quality, calculated as ASR_{fin} $\times$ (QP/Reviewed).	169
8.4	Representative entropy-controlled prompts across three domains: creative writing, education, and journalism. Entropy increases by adding specificity, progressively constraining the response space.	182

8.5	Summary statistics for unwatermarked and watermarked text across different watermarking schemes, highlighting detection scores, automatic quality metrics, and runtime statistics.	183
8.6	Performance metrics for each perturbation oracle. The columns report the number of attack steps, average edits per step, average text perplexity (PPL), average number of grammar errors, average Q approval rate, average rate at which Q blocks every perturbation for a given prompt, average InternLM quality score (QScore), and average runtime per perturbation step in seconds. Emboldened values denote the best performance per metric.	185
8.7	Summary of failed distinguisher tests on the most challenging settings. Classification is first performed by Llama-3.1-70B, followed by GPT-4o on its failures, then o3-mini-high on any remaining cases. The overall 100% success rate indicates that the attacked texts never lose memory of their starting points, contradicting KA1 and suggesting that a stationary distribution is not reached in practice.	194
8.8	Domain distribution for tests which Llama-3.1-70B failed to distinguish.	194
8.9	Entropy distribution for tests which Llama-3.1-70B failed to distinguish.	194
8.10	Distribution of human quality assessments by split for the Sandcastles dataset. The table details counts for cases where attacked outputs were rated as "Attacked Better" or "Tie" (grouped under Quality Preserved (QP)) versus "Original Better", along with overall totals for both training and test sets.	196
8.11	Overview of oracle performance on our human-annotated test set. For each oracle we report average inference time, Quality-Preserved (QP) Precision, Overall F1, and RewardBench (RB) Score when available. Despite fine-tuning on human judgements, no oracle perfectly capture human quality assessments, and high RB Scores did not predict strong performance in our evaluation setting.	197
8.12	Comparison of human agreement rates on quality preservation (QP) percentages between DiffOracle and InternLM. InternLM shows a higher agreement rate, suggesting it aligns better with human judgments.	198

8.13 Attack success rates (ASR) and autoamated quality scores across different perturbation strategies. Human review reveals an average of 59.52% of successfully attacked texts have degraded quality. μ_{w_0} represents the initial watermark score at step 0, while μ_{w_t} represents the final watermark score after t mutation steps. “min” refers to the point where the watermark score is at its lowest during the attack while “fin” refers to score at the final step of the attack. "Reviewed" indicates the number of human-reviewed examples where the watermark was broken. **QP** and $\neg$ **QP** represent the number of cases where human reviewers judged the attacked text as quality-preserving or degraded, respectively. **Q-ASR_{fin}** is the re-estimated attack success after controlling for quality, calculated as $\text{ASR}_{\text{fin}} \times (\text{QP}/\text{Reviewed})$. The remaining quality columns show, for time step 0 and the final step t , InternLM quality score (q), perplexity (p), grammar errors (g), and unique bigram diversity (d). On average, quality degraded significantly while perplexity, grammar errors, and diversity improved. 201

VITA

2017–2019 M.S. in Computer Science
University of California, Los Angeles

2008–2012 B.S. in Business Administration
University of Florida
Majors: Management; Information Systems and Operations Management
Minors: Entrepreneurship; Finance
Thesis: *The Viability of the Student-Run Housing Cooperative: A Case Study of the Cooperative Living Organization*

PUBLICATIONS

Harel-Canada, Fabrice, Wang, Lingxiao, Gulzar, Muhammad Ali, Gu, Quanquan, & Kim, Miryung, Is neuron coverage a meaningful measure for testing deep neural networks?, ESEC/FSE, 2020

Harel-Canada, Fabrice, Gulzar, Muhammad Ali, Peng, Nanyun, & Kim, Miryung, Sibylvariant Transformations for Robust Text Classification, ACL Findings, 2022

Xu, Guangxuan, Liu, Ruibo, **Harel-Canada, Fabrice**, Chandra, Nischal Reddy, & Peng, Nanyun, EnDex: Evaluation of Dialogue Engagingness at Scale, EMNLP Findings, 2022

Kang, Hong Jin*, **Harel-Canada, Fabrice***, Gulzar, Muhammad Ali, Peng, Nanyun, & Kim, Miryung, Human-in-the-Loop Synthetic Text Data Inspection with Provenance Tracking, NAACL Findings, 2024

Harel-Canada, Fabrice Y., Zhou, Hanyu, Muppalla, Sreya, Yildiz, Zeynep Senahan, Kim, Miryung, Sahai, Amit, & Peng, Nanyun, Measuring Psychological Depth in Language Models, EMNLP, 2024, **Outstanding Paper Award (Top 0.25% of submissions)**

Harel-Canada, Fabrice Y., Salimian, Anabel, Moghanian, Brandon, Clingan, Sarah, Nguyen, Allan, Avra, Tucker, Poimboeuf, Michelle, Romero, Ruby, Funnell, Arthur, Petousis, Panayiotis, Shin, Michael, Peng, Nanyun, Shover, Chelsea L., & Goodman-Meza, David, Enhancing Substance Use Detection in Clinical Notes with Large Language Models, Research Square (Preprint), 2025, In Review @ Journal of Drug and Alcohol Dependence

Harel-Canada, Fabrice Y.*, Erol, Boran*, Choi, Connor, Liu, Jason, Song, Gary Jiarui, Peng, Nanyun, & Sahai, Amit, Sandcastles in the Storm: Revisiting the (Im)possibility of Strong Watermarking, ACL, 2025

*Denotes equal contribution

CHAPTER 1

Introduction

1.1 Research Problem

The rapid evolution of artificial intelligence (AI), particularly in machine learning (ML) and language models (LMs), has transformed how we generate, process, and interpret data. Yet as these systems grow more sophisticated, their evaluation frameworks struggle to keep pace. Traditional metrics – often borrowed from software engineering or statistical baselines – focus narrowly on structural or quantitative properties, neglecting the nuanced interplay of diversity, human judgment, and subjective quality that defines real-world utility. This dissertation argues that advancing AI evaluation requires a paradigm shift: from rigid, *structural* metrics toward holistic frameworks that harmonize *data diversity* and *human-centric subjectivity*.

Extensive progress in ML has already led to integration in safety-critical domains [26, 132]. However, *all* software, including ML-based software, is prone to unanticipated defects. In traditional software engineering, defects typically arise from algorithmic design or flawed implementation. By contrast, ML-integrated systems add several additional, fault-prone dimensions, especially pertaining to the dataset itself. Commonly cited are *data bugs* [122], including insufficiently *diverse* data (low input coverage) or noisy data with misaligned labels. To complicate matters further, it is unclear whether existing test metrics (e.g., code coverage) can be adapted to ML, where high-dimensional feature spaces and model complexity can render naive coverage measures (like neuron coverage) uninformative or even misleading.

As ML expands into natural language processing and generation, evaluation challenges intensify. Beyond correctness, we often care about how *engaging*, *informative*, or *enjoyable* the outputs are from a human standpoint – qualities left unaddressed by purely structural approaches.

Meanwhile, as sophisticated language models blur the line between machine-generated and human-written text, *authorship verification* (e.g., watermarking) emerges as a critical safeguard.

The main thesis of this work is that robust and meaningful AI evaluation must consider data diversity, subjective reader experience, and techniques for verifying authorship. By accounting for these factors, we can develop a more comprehensive and human-centric framework for assessing and improving ML models.

Concretely, this dissertation addresses the following subhypotheses, which collectively redefine ML evaluation to be more holistic:

- (RQ1) *How effectively can traditional structural coverage metrics (e.g., neuron coverage) be adapted to the unique challenges of testing deep neural networks?*
- (RQ2) *How can data transformations be reformulated to expand data-space coverage and improve model performance on generalization, defect detection, and robustness?*
- (RQ3) *How can we leverage feature-aware augmentation strategies and human-in-the-loop inspection to balance diversity with text quality, reducing noise while amplifying beneficial transformations?*
- (RQ4) *Which subjective qualities (e.g., empathy, narrative complexity) matter for human-centric evaluation of language model outputs, and how can we systematically measure them?*
- (RQ5) *Can watermarking techniques remain viable for verifying AI authorship in the face of increasingly powerful generative models and adversarial attacks?*

If successful, the work in this dissertation will show that “more coverage” is not enough; AI systems must be measured against richer concepts of diversity and human-centric values. By infusing these dimensions into ML testing and inspection, we aim to improve both technical robustness and real-world utility.

1.2 Is Neuron Coverage a Meaningful Metric for Testing Deep Neural Networks?

To systematically increase NC during test generation, we develop a novel *diversity-promoting regularizer* that can be plugged into existing adversarial attack algorithms such as PGD [199] and CW [39]. This regularizer penalizes skewed layer-wise activations to promote more diverse neuron activation distributions. As a result, our regularizer can be added to augment existing adversarial attack methods so that these methods can induce previously inactive neurons to fire and thereby increase NC. While prior work [238, 296] has attempted to improve a few neurons’ activation magnitudes at each optimization step, our diversity-promoting regularizer makes this process more systematic by incorporating NC increase and diversification into the optimization objective.

We then assess the generated test suites using three criteria. The first is *defect detection* capability, i.e., the ability to detect adversarial attacks. The second is the *naturalness* of the generated test inputs and we use the Inception Score (IS) [268, 19] and the Fr chet Inception Distance (FID) [115, 216] to assess how realistic the generated test inputs are. The third criterion is *output impartiality*, the degree to which model predictions are biased (or unbiased) towards particular class labels. Assessing impartiality is inspired by the output-uniqueness test selection criteria [2], as the test suite must exercise *diverse* output behavior and should not prefer only a few output values. We quantify output impartiality via Pielou’s evenness [243], an *entropy-based* measure [274] from the field of ecology.

Equipped with the above evaluation metrics and the novel diversity promoting input generation method, we investigate the trade-offs between neuron coverage, defect detection, naturalness and output impartiality. We study two image classification datasets (MNIST and CIFAR10), one autonomous vehicle dataset (Udacity Self-Driving Car), six classification-based DNN models, two regression-based DNN models, and two attack algorithms (CW and PGD). In total, 2095 test suites, over 200,000 images, are generated. Each test suite represents a different configuration of models, datasets, attack algorithms, and hyperparameter combinations used for targeting certain layers and promoting diversity in neuron activations. This extensive analysis finds that *increasing NC actually makes it harder to generate an effective, diverse test suite*.

Overall, our findings invoke skepticism that neuron coverage may not be a meaningful measure for testing deep neural networks. This result is aligned with recent skepticism that, while code coverage remains a widely used test adequacy criterion [126, 28], code coverage may not be correlated with defect detection [125] and thus may not be a meaningful metric by itself. These findings call for a new test generation method that not only improves defect detection, but also promotes naturalness and output impartiality to create *realistic* inputs and to exercise *diverse* output behavior.

Testing of traditional software leverages code coverage, or the unique lines of code executed by a test, to implicitly guide the generation of diverse test inputs. However, it is not clear that such *structural* coverage metrics are meaningful when adapted for ML models. We hypothesized for (RQ1) that structural coverages, like neuron coverage (NC), are not yet informative enough to help generate diverse test data. We evaluated this hypothesis by inducing higher NC via a novel neural activation diversity-promoting technique and measured the correction of NC against defect detection, naturalness, and output impartiality (i.e. output diversity). These results confirm the hypothesis that structural coverage metrics are not meaningful in isolation and that test objectives should consider diversity objectives like output impartiality.

1.3 Data Augmentation and its Impact on Data Diversity and Model Performance

1.3.1 Sibylvariant Transformations for Robust Text Classification

Automatically generating new data is a critical component within modern machine learning pipelines. During training, data augmentation can expose models to a larger, more diverse portion of the potential input space, consistently leading to better generalization and performance [278, 156, 239]. After training, creating diverse test instances from existing data can expose specific model failure modes and provide actionable corrective feedback [342, 262].

While many techniques can artificially expand labeled training sets or test suites, nearly all of

them are class-preserving. That is to say, the model outputs are invariant (INV) with respect to the transformations. This cautious constraint ensures the new data does not lie in an out-of-distribution null class that might impede the learning objective. However, it also requires more conservative transforms that inherently limit the degree of diversification.

In this work, we propose and extensively investigate the potential of *sibylvariant* (SIB) transformations that *knowably* vary the expected class. From the Greek *sibyls*, or oracles, the term parallels the oracle construction problem in software testing [18]. In a nutshell, sibylvariants either fully transmute a datum from one class c_i to another c_j , or mix data from multiple classes together to derive a new input with a soft label that reflects the mixed membership. In this way, sibylvariance can more strongly perturb and diversify the underlying distribution. Moreover, SIB makes possible a new type of adaptive training by synthesizing data from frequently confused class pairs, challenging the model to differentiate with greater refinement.

Our comprehensive evaluation encompasses six text classification datasets, 11 transformation pipelines, and three different levels of data availability. In total, we trained 216 models and generated over 30 million new training inputs, 480,000 testing inputs, and 3,300 adversarial inputs. In our generalization study, SIB attained the highest accuracies in 89% (16 out of 18) of experimental configurations, with the adaptive mixture mutations being the most consistently effective. SIB transforms also revealed the greatest number of model defects in 83% (5 out of 6) of the testing configurations. Lastly, of all the experimental configurations where adversarial robustness was improved over the no-transform baseline, 92% (11 out of 12) of them involved SIB. Overall, our findings strongly support the efficacy of sibylvariance for generalization performance, defect detection, and adversarial robustness.

Within software engineering and ML, data transformations are ubiquitously used to expand test and training sets and improve downstream performance metrics like model accuracy. The vast majority of these transformations employ small perturbations to preserve the underlying semantics of the original data and are therefore inherently limited in the amount of input coverage they can reach. Our hypothesis for **(RQ2)** was that we can drop the label-preserving constraint and jointly transform the input and output label together to significantly expand data-space diversity. This expanded coverage was then observed to have a positive impact on downstream objectives like improving model generalization after training, and better defect detection and robustness during testing. These results were empirically studied primarily in the context of natural language processing (NLP), but the idea of sibylvariance can be used to diversify data in any domain (e.g. images, time series, tabular data, etc).

While sibylvariance broadens input coverage and boosts robustness, yet nearly all data augmentation introduces some degree of noisy quality degradation. Addressing mislabeled or low-quality text requires finer-grained control over transformations and the ability to filter flawed outputs. The following works, FADA and INSPECTOR, tackle these challenges by combining text-quality metrics and semantic insights (FADA) with human-in-the-loop oversight (INSPECTOR) to maintain high-quality augmented data.

1.3.2 FADA: Learning Effective Data Augmentation Policies with Text Quality Metrics and Semantic Features

Data augmentation has become indispensable for improving model generalization [83, 276], particularly in low-resource NLP settings. However, the proliferation of augmentation techniques – from synonym substitution to back-translation – creates a scalability challenge: practitioners face a combinatorial explosion of choices with limited guidance on which transformations will improve quality or diversity for their specific task. Prior work often evaluates augmentation strategies post hoc by training models on augmented data and measuring downstream performance. While effective, this approach is computationally prohibitive for large search spaces.

To address this, we propose *feature-aware data augmentation (FADA)*, a framework that jointly optimizes text quality metrics (e.g., label alignment, fluency, grammaticality) and semantic feature targeting. FADA extracts abstract meaning representations (AMRs) from text to identify high-level features, such as negations, co-references, and named entities. By monitoring how transformations interact with these features, FADA learns augmentation policies that maximize diversity while preserving critical semantics. For example, `RandomDeletion` is selectively applied to non-negated sentences to avoid flipping sentiment labels, while `Concept2Sentence` – a novel transform that replaces entities with descriptive clauses – targets texts lacking contextual richness.

Experiments across four text classification tasks show that FADA improves low-resource accuracy by 15% over baseline augmentations while maintaining 92% of original text quality. Surprisingly, text quality metrics like fluency (perplexity) and label alignment (confident learning scores) initially correlated with post-training performance ($r = 0.68$) but became decoupled after policy optimization. This suggests that while quality metrics guide initial policy design, feature-aware diversification drives final gains. FADA’s policies also transfer across domains: a policy trained on sentiment analysis improved hate speech detection accuracy by 9% without fine-tuning.

(RQ3) posits that diversity and provenance can enhance augmentation and debugging. FADA operationalizes this by linking transformations to semantic features (provenance) and optimizing for diversity-quality trade-offs. By demonstrating that feature-aware policies improve generalization and transferability, this work establishes a blueprint for data augmentation that is both systematic and interpretable.

1.3.3 Human-in-the-Loop Synthetic Text Data Inspection with Provenance Tracking (INSPECTOR)

Data augmentation inherently introduces noise: transformations may distort semantics, degrade fluency, or misalign labels. While tools like FADA mitigate these risks algorithmically, human validation remains critical for high-stakes applications. However, inspecting thousands of synthetic texts is labor-intensive. Prior work attempts to automate quality control via LLM-based filters, but these often inherit model biases or miss subtle errors.

INSPECTOR addresses this by combining *transformation provenance* (tracking which augmentations generated each text) and *assistive labeling* (leveraging LLMs to flag potential label errors). Users group texts by shared transformations (e.g., all outputs from `WordSwap`) or linguistic features (e.g., negations), then review aggregated statistics like label consistency and fluency scores. For example, INSPECTOR revealed that `SentMix` – a sibylvariant transform blending sentences from two classes – produced 23% mislabeled outputs on a hate speech dataset, disproportionately affecting texts with sarcasm.

In a user study, INSPECTOR tripled the rate of identifying mislabeled texts compared to unaided inspection. Participants using provenance filtering corrected 4x more labels in hate speech detection tasks, with 89% reporting higher confidence in their decisions. Notably, LLM-assisted labeling alone underperformed human-AI collaboration: automated suggestions had 77% accuracy, but users achieved 92% accuracy when guided by provenance groups. This underscores the necessity of human oversight even in augmented pipelines.

(RQ3) emphasizes leveraging provenance for inspection. INSPECTOR directly answers this by demonstrating that transformation-aware grouping reduces human effort while increasing precision. By linking synthetic texts to their generative origins, INSPECTOR enables targeted debugging of augmentation strategies – for instance, identifying that `RandomInsertion` harms coherence in long-form narratives. This work bridges the gap between automated augmentation and human accountability, a prerequisite for deploying ML systems in sensitive domains.

1.4 Measuring Psychological Depth in Language Models (PDS)

While diversity and provenance address *quantitative* aspects of ML evaluation, subjective reader experiences – empathy, engagement, emotional resonance – define the *qualitative* impact of AI-generated narratives. Existing metrics like BLEU [235] or BERTScore [345] fail to capture these dimensions, creating a mismatch between algorithmic performance and human perception.

The Psychological Depth Scale (PDS) addresses this by evaluating five components: (1) empa-

thy (reader’s connection to characters), (2) engagement (narrative pull), (3) emotion provocation (intensity of feeling), (4) authenticity (perceived human-likeness), and (5) narrative complexity (structural richness). Human evaluators achieved strong agreement (Krippendorff’s $\alpha = 0.72$) when rating 97 stories from GPT-4, Llama-3, and human authors. Surprisingly, GPT-4 surpassed human-authored Reddit stories in empathy (Cohen’s $d = 0.41$) and complexity ($d = 0.38$), with 73% of its outputs misclassified as human-written. Automated evaluation via Mixture-of-Personas (MoP) prompting correlated with human judgments ($\rho = 0.51$), though constrained decoding on Llama-3 achieved higher alignment ($\rho = 0.68$).

Subjective user experience is pivotal for evaluating how readers perceive machine-generated narratives. The PDS operationalizes this by assessing empathy, engagement, emotion, authenticity, and narrative complexity with high inter-rater reliability. Empirically, GPT-4 [228] rivaled or surpassed human-authored texts in multiple PDS dimensions, and automated scoring methods (e.g., MoP prompting) correlated well with human judgments. These findings underscore the necessity of going beyond structural metrics to capture *qualitative* reader impact – while simultaneously posing new challenges for distinguishing human from machine-generated content. When machines excel at "human" qualities like empathy, traditional markers of creativity become unreliable signals of provenance. This duality underscores the urgency of robust watermarking, as explored next.

1.5 Sandcastles in the Storm: Revisiting the (Im)Possibility of Strong Watermarking

Watermarking – embedding detectable signals in AI-generated text – is a cornerstone of ethical AI provenance. However, recent theoretical work [341] argues that watermarking is inherently breakable via perturbation attacks that preserve text quality. These attacks assume (1) rapid mixing (watermarks dissolve after minor edits) and (2) perfect quality oracles (automated tools that guide perturbations without degrading output).

Key Assumption 1 (Rapid Mixing): Theoretically, perturbations should erase watermarks by

driving texts toward a stationary distribution independent of their origin. However, our experiments with three watermarking schemes - KGW (statistical hashing) [147], SIR (semantic invariance) [182], and Adaptive (entropy-based) [186] - reveal starkly different behavior. Even after *hundreds of iterative perturbations* (word swaps, paraphrasing, syntax alterations), 100% of texts retained detectable traces of their watermarked origins. For example, Adaptive watermarking maintained persistent signals in low-entropy regions (e.g., semantically stable clauses identified through dependency parsing), requiring destructive edits to remove – a practical impossibility for adversaries seeking to preserve meaning.

Key Assumption 2 (Perfect Oracles): Attacks require automated quality oracles to guide perturbations without degrading output. Yet state-of-the-art LLM-based oracles achieved only 77.4% F1-score in distinguishing quality-preserving edits, with errors compounding over attack iterations. Human validation further exposed this brittleness: while automated metrics labeled 26.1% of attacks as successful, human reviewers tasked with determining whether quality was preserved dropped the success rate to 10%.

(RQ5) highlights the role of provenance in trustworthy AI. By demonstrating watermarking’s practical robustness, this work reinforces provenance as a viable mechanism for accountability. The gap between theoretical attacks and real-world constraints – slow mixing, imperfect oracles – suggests watermarking remains a critical tool for tracing AI-generated content.

1.6 A Unified Framework for Evaluation

Collectively, these works chart a holistic path for evaluating and improving ML systems. From early revelations on **structural limitations** (neuron coverage) to sophisticated data augmentation methods (sibylvariance, FADA), from **human-in-the-loop inspection** (INSPECTOR) to **subjective reader-based metrics** (PDS), and finally to **authorship verification** (watermarking), this dissertation builds toward a paradigm where:

- **Structural Coverage Is Not Enough.** Neuron coverage rarely correlates with real defect

detection, data naturalness, or output diversity.

- **Data-Centric Perspectives Boost Robustness.** Label-shifted augmentations (SIB) and feature-aware policies (FADA) systematically broaden input diversity.
- **Human Judgment Complements Automation.** INSPECTOR highlights the challenge of maintaining quality during data diversification, and the importance of human intervention to filter out low-quality augmentations.
- **Subjective and Experiential Factors Matter.** The Psychological Depth Scale reminds us that user-centric experiences – engagement, empathy, authenticity – are critical for certain AI applications.
- **Authorship Verification Remains Possible.** Watermarking is not trivial to defeat in practice, underscoring the feasibility of provenance tracking in an era of near-human-level text generation.

In sum, modern AI requires evaluation methods that unify the best of *structural rigor*, *data coverage*, *human judgment*, and *authorship verification* techniques. By transcending static, reductionist metrics, we can align ML systems more closely with real-world needs: reliability, human satisfaction, creative expression, and accountability.

Roadmap. The subsequent chapters expand on these themes:

- **Chapter 2: Structural Limitations.** We delve into the shortcomings of neuron coverage for deep neural networks.
- **Chapters 3–5: Data Augmentation and Human-in-the-Loop.** We detail sibylvariance and feature-aware techniques, then demonstrate the role of human oversight via INSPECTOR.
- **Chapters 6–7: Subjective Evaluation and Authorship Verification.** We propose the Psychological Depth Scale to capture subjective impact, then address the viability of watermarking to verify authorship as human and AI writing quality converge.

Overall, this dissertation showcases how a deeper, human-centric understanding of diversity, quality, subjective experience, and provenance can elevate the testing and evaluation of AI systems.

CHAPTER 2

Background

The rapid evolution of deep learning systems has spurred complementary research threads in testing methodologies, data augmentation, model evaluation, creative AI assessment, and content provenance verification. We synthesize these interconnected domains below, emphasizing their theoretical foundations and practical limitations.

2.1 Testing and Coverage Metrics for Deep Learning

The testing of deep neural networks (DNNs) has emerged as a critical frontier in ensuring the reliability of AI systems, driven by their widespread adoption in safety-critical domains such as autonomous vehicles and healthcare [116, 157, 277]. Early efforts drew inspiration from traditional software testing paradigms, adapting concepts like differential testing [206] and metamorphic testing [272] to address the unique challenges posed by DNNs’ data-driven, non-linear architectures. Unlike deterministic software, DNNs derive behavior from learned statistical patterns rather than explicit rules, rendering conventional coverage metrics (e.g., code coverage) inadequate. This gap catalyzed the development of structural coverage metrics tailored to DNNs, with neuron coverage (NC) [238] – the proportion of activated neurons under a test suite – emerging as a pioneering heuristic.

Initial studies posited NC as a proxy for input diversity, arguing that higher activation rates correlate with broader exploration of a model’s decision logic. For instance, DeepXplore [238] demonstrated that maximizing NC via gradient-guided input perturbations could expose erroneous behaviors in image classifiers, outperforming random testing. Subsequent work, such as DeepGauge [194], expanded NC into a multi-granular framework, introducing layer-level criteria (e.g.,

boundary coverage) to account for the hierarchical nature of DNNs. These metrics were further operationalized in regression testing [194] and adversarial robustness evaluation [100], where adversarial examples were shown to increase NC while degrading model accuracy.

However, the validity of NC as a meaningful adequacy metric has faced mounting scrutiny (both in the present work and others [327, 337]). Recent empirical studies [76, 177] reveal weak or inconsistent correlations between NC and model robustness, particularly when test suites prioritize NC maximization over semantic validity. For example, [273] demonstrated that NC-guided adversarial examples often produce unrealistic inputs (e.g., distorted images), inflating coverage without improving defect detection. This aligns with critiques that NC reduces neurons to binary activation states, ignoring the semantic context of their activations [142]. Consequently, newer metrics like surprise adequacy [142] – which quantifies the "unexpectedness" of activations relative to training data – and uncertainty-aware coverage [195] – which incorporates predictive confidence – have gained traction as more holistic alternatives.

The interplay between testing and adversarial attacks further complicates this landscape. While adversarial methods [199] remain a cornerstone of DNN testing, their efficacy hinges on perturbation constraints (e.g., ℓ_p -norms) that balance realism and detectability. Domain-specific transformations, such as GAN-based style transfers [344] or weather simulations for autonomous driving [296], offer richer semantic diversity but risk overfitting to synthetic artifacts. This tension underscores a fundamental challenge: coverage metrics must reflect both structural diversity and real-world utility.

2.2 Data Augmentation and Transformation Paradigms

Data augmentation has long served as a cornerstone for improving model generalization, but its evolution reflects a deepening understanding of how and why transformations impact learning [217]. Early approaches focused on label-preserving invariant (INV) transformations/modifications that retain semantic meaning (e.g., image rotations, text synonym swaps) to reinforce model invariances [156, 278]. These methods implicitly assume that robustness to local perturbations (e.g., lighting changes, typos) translates to broader generalization, a hypothesis validated by their success in

vision [350] and NLP [83]. However, INV techniques face inherent limitations: they expand the support of the training data only marginally, risking overfitting to “augmentation artifacts” [324], and struggle to address class imbalance or underrepresentation of edge cases.

The emergence of what we have called sibylvariant (SIB) transforms – alterations that deliberately modify semantic labels – marked a paradigm shift. By synthesizing mixed or transmuted samples (e.g., MixUp’s linear interpolations [340], CutMix’s patch replacements [338]), SIB methods explicitly broaden the training distribution’s support, forcing models to learn smoother decision boundaries. In NLP, techniques like SentMix [105] blend sentence pairs to simulate compositional reasoning, while style-transfer methods [354] decouple content from form to encourage invariant feature extraction. Crucially, SIB transforms reject the INV assumption of strict label preservation, instead embracing controlled semantic shifts that mirror real-world ambiguity (e.g., a partially occluded stop sign interpreted as “yield”).

Automated augmentation policies further refined this paradigm by dynamically tailoring transformations to model needs. Works like AutoAugment [62] and Population-Based Augmentation [117] used reinforcement learning to discover optimal INV policies for vision tasks, while text-focused approaches [222] adapted these ideas to lexical perturbations. However, these methods often prioritize quantity over quality, generating implausible samples that degrade robustness [241]. Recent advances address this via quality-aware policies that jointly optimize linguistic metrics (e.g., grammar correctness, semantic similarity) and diversity [179], though computational costs remain prohibitive for large models.

A critical yet understudied challenge lies in the text-vs.-vision dichotomy. While image transforms exploit continuous pixel spaces for smooth interpolations [72], text’s discrete, compositional nature demands structure-aware perturbations. Methods like back-translation [318] and token-level swaps [83] approximate this but risk incoherence. Generative approaches, such as GPT-based paraphrasing [324] or diffusion models [329], offer richer diversity but introduce distributional drift: synthetic texts that diverge from human patterns. This tension underscores a fundamental trade-off: augmentations must perturb sufficiently to explore underlearned regions while preserving naturalness to avoid adversarial overfitting.

The efficacy of SIB transforms hinges on their semantic plausibility – a poorly executed mixup may teach spurious correlations (e.g., blending "positive" and "negative" reviews creates uninterpretable labels). Similarly, automated policies risk shortcut learning, favoring trivial transformations (e.g., punctuation changes) over semantically meaningful ones [55]. In the following section, we discuss the potential of integrating human-in-the-loop curation [41] and foundation model guidance [1] to align synthetic data with human intuition, bridging the gap between algorithmic diversity and pragmatic utility.

2.3 Model Debugging and Synthetic Data Inspection

The proliferation of synthetic data in modern AI pipelines – driven by generative models [1, 153] and automated augmentation [62] – has intensified the need for robust debugging tools that address both model errors and data quality. Traditional model-centric approaches, such as DynaBench [140] and AdaTest [260], focus on identifying failure modes by generating challenging test cases or refining user-provided examples. While effective for diagnosing model biases (e.g., over-reliance on spurious correlations), these tools often neglect the provenance of training data a critical gap when synthetic or augmented data introduces latent artifacts or distributional shifts.

To address this, we proposed INSPECTOR, a data-centric debugging framework that shifts the locus of analysis from model behavior to data lineage. By tracking the provenance of synthetic texts including augmentation histories, generative model versions, and transformation parameters INSPECTOR enables users to identify systemic flaws in data pipelines (e.g., overuse of certain paraphrasing templates [324] or adversarial perturbations [199]). Unlike tabular data tools like Data Civilizer [259], which debug pipeline errors via breakpoints and constraint violations, INSPECTOR adapts to text’s unstructured nature through linguistic feature clustering. For instance, grouping texts by AMR nodes (abstract meaning representation [14]) reveals unintended stylistic biases introduced during augmentation.

A key innovation lies in balancing automation with human oversight. While fully automated methods like Zeno’s metric-driven slicing [35] or Tempura’s template abstraction [323] risk missing nuanced errors (e.g., subtly nonsensical sentences from GPT-4 [1]), INSPECTOR bridges this

divide via proactive filtering: surfacing suspicious samples (e.g., low-grammaticality texts [241]) while allowing users to iteratively refine filters based on domain-specific priorities. This advances prior interactive paradigms like Wrangler [133] by replacing hand-coded constraints with learned linguistic heuristics (e.g., sentiment coherence [237]).

INSPECTOR also addresses challenges in synthetic data ecosystems, where models trained on other models’ outputs risk “cascading hallucinations” [328]. By correlating data lineages with model confidence metrics, INSPECTOR flags samples generated by low-temperature sampling (prone to repetition [1]) or heavily mutated via SIB transforms, which may exhibit label ambiguity. This extends token-based rule learning [53] to feature-space patterns, leveraging embeddings to detect distributional outliers.

2.4 Creativity Assessment in Generative AI

Evaluating creativity in machine-generated narratives poses unique challenges at the intersection of literary theory, cognitive psychology, and computational metrics [10]. Traditional methodologies like the Consensual Assessment Technique (CAT), which relies on expert judgments to assess creativity as a socially constructed trait [4], provide a human-centric foundation rooted in psychometrics. However, CAT’s reliance on curated panels of experts becomes impractical for AI-generated content produced at scale. While automated metrics like the Torrance Test of Creative Writing (TTCW) [304] and LLM-as-judge frameworks [139] offer scalability, they risk oversimplifying psychological depth by reducing creativity to quantifiable surface features such as word rarity or syntactic diversity. For instance, TTCW-based studies concluded that humans outperform LLMs by 3-10× in creativity [41], but their reliance on lengthy, formulaic prompts and fixed-length narratives may underestimate the nuanced potential of AI-generated text.

Theoretical foundations from literary criticism, such as reader-response theory [84, 202] and text-world theory [89], emphasize the reader’s dynamic role in constructing meaning – a lens critical for evaluating AI-generated content consumed by diverse audiences. Empirical studies converge on five dimensions of psychological depth: emotion provocation, empathy, engagement, authenticity, and narrative complexity [58, 225, 93, 220, 150]. Yet, controlled studies reveal persistent

gaps in AI-generated narratives, particularly in authenticity and coherence, even as co-writing systems like Wordcraft [336] and specialized architectures like Weaver [312] demonstrate LLMs’ potential to enhance human creativity through open-ended collaboration. For example, while LLMs excel at generating grammatically fluent text, they often produce stories that lack the lived-in texture of human experiences or falter in long-form narrative planning [328].

These challenges highlight the need for evaluation frameworks that reconcile scalability with psychological validity. A multi-dimensional approach, grounded in universal reader responses rather than expert judgments alone, quantifies creativity through metrics like emotional resonance and attentional engagement while addressing the contextual fluidity of creative expression across cultures [9, 297]. However, as AI systems grow more sophisticated, the ability to distinguish human-written from machine-generated text becomes critical for transparency and equitable evaluation. Watermarking techniques [341, 233] and cryptographic auditing [257] aim to track synthetic content, ensuring accountability in creative domains. Yet, such methods must balance robustness against adversarial attacks [341] with preserving the psychological depth that defines meaningful narratives. This interplay between creativity assessment and content provenance underscores a broader imperative: as AI reshapes creative landscapes, frameworks for evaluation must evolve in tandem, ensuring they capture both technical capability and the human values underpinning artistic expression. The following section explores these tensions through the lens of watermarking, examining how attribution mechanisms can coexist with, and even enhance, efforts to measure the psychological richness of AI-generated works.

2.5 Watermarking and Content Provenance

Text watermarking involves embedding imperceptible, machine-detectable signals into AI-generated content to authenticate its synthetic origin, addressing the growing challenge of distinguishing machine-authored text from human work as LLMs achieve human-like fluency [183]. This capability is critical for combating misinformation, plagiarism, and intellectual property disputes, particularly as AI-generated news articles, social media posts, and academic abstracts proliferate [46]. Watermarking enables authorship provenanceverifying whether content originated from a

specific model or user while preserving utility and naturalness [191]. However, designing effective schemes requires balancing competing priorities: strong watermarks are easily detected but risk distorting semantics, while weak ones evade detection but are vulnerable to removal, exemplifying the detectability-robustness trade-off [266].

One common approach, lexical watermarking, alters token selection during generation by biasing the model’s output distribution. For example, the KGW scheme [147, 148] partitions the vocabulary into "green" and "red" tokens using a hash function, favoring green tokens during sampling. Detection relies on statistically significant green-token bias, but such methods are vulnerable to paraphrasing attacks that preserve meaning while redistributing tokens [153]. In contrast, semantic watermarking embeds signals in the text’s meaning rather than its surface form. Techniques like SIR [182] and Adaptive watermarking [186] incorporate semantic embeddings in their selective vocab boosting, adjusting watermark strength based on context (e.g., stronger signals in high-entropy creative text).

One prominent analysis frames watermark removal as a random walk over the space of quality-preserving text mutations [341], where adversaries iteratively apply perturbation oracles (e.g., semantic-preserving rewrites [153]) to erase watermarks. This process is governed by the mixing time of the induced graph, which quantifies the steps needed for perturbations to converge to a stationary distribution independent of the original watermark. For high-entropy prompts (e.g., open-ended stories), the exponential state space renders attacks computationally prohibitive [254].

CHAPTER 3

Is Neuron Coverage a Meaningful Metric for Testing Deep Learning Networks?

Testing of traditional software leverages code coverage, or the unique lines of code executed by a test, to implicitly guide the generation of diverse datasets.¹ However, it is not clear that such *structural* coverage metrics are meaningful when adapted for ML models. We hypothesize for **(RQ1)** that structural coverages, like neuron coverage (NC), are not yet informative enough to help generate diverse test data. We evaluated this hypothesis by inducing higher NC via a novel neural activation diversity-promoting technique and measured the correction of NC against defect detection, naturalness, and output impartiality (i.e., output diversity).²

3.1 Introduction

Extensive progress in machine learning has enabled computers to model expected behavior with minimal human guidance and has led to its integration into many safety-critical systems [26, 132]. Since all software is prone to unanticipated and undesirable defects, creating test suites and assessing their quality is an important part of building confidence during the software lifecycle.

To assess the test adequacy of neural networks, prior work proposed neuron coverage (NC) [238] and its variants [296, 194]. This notion of NC builds on the intuition of code coverage, whilst recognizing the unique challenges and structures of neural networks. NC describes the proportion of neurons activated beyond a given threshold. The intuition here is that NC captures the magnitude

¹In general, coverage is closely related to diversity. Some prior work uses them interchangeably [275], though we emphasize that coverage does not consider type frequency or intra-type similarity.

²This work was completed between the master’s and PhD programs at UCLA.

of individual neuron activations independently and thus serves as a proxy for observing model behavior. Based on the implicit assumption that increasing NC can improve test suite quality, NC was used to guide test generation [238, 296]. Prior work found preliminary evidence that NC is correlated with defect detection capability [296].

To systematically increase NC during test generation, we develop a novel *diversity-promoting regularizer* that can be plugged into existing adversarial attack algorithms such as PGD [199] and CW [39]. This regularizer penalizes skewed layer-wise activations to promote more diverse neuron activation distributions. As a result, our regularizer can be added to augment existing adversarial attack methods so that these methods can induce previously inactive neurons to fire and thereby increase NC. While prior work [238, 296] has attempted to improve a few neurons’ activation magnitudes at each optimization step, our diversity-promoting regularizer makes this process more systematic by incorporating NC increase and diversification into the optimization objective.

We then assess the generated test suites using three criteria. The first is *defect detection* capability, i.e., the ability to detect adversarial attacks. The second is the *naturalness* of the generated test inputs and we use the Inception Score (IS) [268, 19] and the Fr chet Inception Distance (FID) [115, 216] to assess how realistic the generated test inputs are. The third criterion is *output impartiality*, the degree to which model predictions are biased (or unbiased) towards particular class labels. Assessing impartiality is inspired by the output-uniqueness test selection criteria [2], as the test suite must exercise diverse output behavior and should not prefer only a few output values. We quantify output impartiality via Pielou’s evenness [243], an *entropy-based* measure [274] from the field of ecology.

Equipped with the above evaluation metrics and the novel diversity promoting input generation method, we investigate the trade-offs between neuron coverage, defect detection, naturalness and output impartiality. We study two image classification datasets (MNIST and CIFAR10), one autonomous vehicle dataset (Udacity Self-Driving Car), six classification-based DNN models, two regression-based DNN models, and two attack algorithms (CW and PGD). In total, 2095 test suites, over 200,000 images, are generated. Each test suite represents a different configuration of models, datasets, attack algorithms, and hyperparameter combinations used for targeting certain layers and promoting diversity in neuron activations. This extensive analysis finds that increasing NC actually

makes it harder to generate an effective test suite.

1. **Defect Detection:** Only 2 out of 64 experimental results supported the hypothesis that NC is both strongly and positively correlated with defect detection (i.e., adversarial attack success), whereas 33 were negatively correlated, implying that increasing NC is likely to harm defect detection.
2. **Naturalness:** Only 1 out of 64 results supported the hypothesis that NC is both strongly and positively correlated with the realism and naturalness of the inputs, whereas 44 were negatively correlated, implying that increasing NC is likely to make the generated inputs more unnatural.
3. **Output Impartiality:** Only 3 out of 64 results supported the hypothesis that NC is both strongly and positively correlated with impartiality in output predictions, whereas 21 were negatively correlated. Certain class labels have higher NC by default and the process of increasing NC in fact biases perturbations towards those output class labels.

Our key contributions are summarized as follows:

- We develop a novel regularization technique that can be seamlessly integrated into existing adversarial attack methods to promote neural activation diversity and increase neuron coverage during test suite generation.
- We adopt the Inception Score (IS) [268] and Fr chet Inception Distance (FID) [115] as generic, scalable, and automatic means of evaluating naturalness. We are the first to apply Pielou’s evenness [243] to examine the previously under-investigated issue of output impartiality in test suites.
- We conduct extensive evaluations to show that NC is neither positively nor strongly correlated with attack success, input realism, and output impartiality, which we argue are important properties to consider when testing DL systems.
- We put forward the complete code and artifacts to automatically generate test suites and replicate our empirical analysis at <https://doi.org/10.5281/zenodo.4021473>

Overall, our findings invoke skepticism that neuron coverage may not be a meaningful measure for testing deep neural networks. This result is aligned with recent skepticism that, while code coverage remains a widely used test adequacy criterion [126, 28], code coverage may not be correlated with defect detection [125] and thus may not be a meaningful metric by itself. Similar to how Inozemseva *et al.* [125] highlight an empirical lack of correlation between traditional code coverage and defect detection, our result is about a lack of *correlation*, not causation. We do not claim that NC is useless; rather, we warn researchers about the potential misuse of NC as *the objective* for test generation because a naive attempt to increase NC could sacrifice other desired properties.

These findings call for a new test generation method that not only improves defect detection, but also promotes naturalness and output impartiality to create *realistic* inputs and to exercise *diverse* output behavior. This argument to incorporate additional objectives is aligned with a recent survey of testing ML-based systems [343] that lists multiple desired testing properties, including correctness, model relevance, robustness, security, efficiency, fairness, interpretability, privacy, and surprise adequacy. Satisfying such multiple objectives may necessitate the use of multi-objective search techniques [165] or enable users to easily add domain-specific constraints to guide meaningful input transformation and oracle checking in metamorphic testing [272].

3.2 Related Work

This section reviews related work on DL systems, DNN testing, and adversarial attacks. Work relevant to our methodology is described in greater detail in Section 3.3.

Deep Learning Systems. DNNs have achieved many breakthroughs in the field of artificial intelligence, such as speech recognition [116], image processing [157], statistical machine translation [12], and game playing [277]. Each DNN contains basic computational units called neurons, which are connected with one another via edges of varying importance or weight. Neurons apply a nonlinear activation function to the inner product of their inputs and weights to output a value, which becomes the input to a subsequent neuron. Layers are used to organize the directed connections between neurons and there is always one or more hidden layers between one input and one output layer. Overall, a DNN can be viewed as a meta-function that aggregates the weighted

contributions from its neural sub-functions to map some input into some target output. Suboptimally set weights make the DL system vulnerable to erroneous behaviors and the opacity of these numerically-derived rules make them difficult to understand and debug.

DNN Testing.

With the success of deep learning, there emerged a line of research into testing DNNs by leveraging the ideas in traditional software testing methods [206, 111]. We discuss several of the most relevant DNN testing methods that utilize the NC-based criteria as follows.

DeepXplore [238] is a white-box differential testing algorithm that leverages NC to guide systematic exploration of DNN’s internal logic. Input images are modified by several domain-specific transformations, and a transformed image is selected for inclusion into a test suite if it fools at least one of several similarly trained DNNs. Their study finds that NC is a better metric than code coverage and increasing NC tends to increase ℓ_1 -distance among inputs.

DeepTest [296] is a gray-box, NC-guided test suite generation approach using metamorphic relations. This effort introduced a wider range of affine transformations to predict the steering angle of an autonomous vehicle. DeepRoad [344] is a GAN-based metamorphic testing approach that utilizes a shared latent space representation to perform a sophisticated style transfer of some target road condition, i.e., rain, snow, etc., to a given source image. DeepRoad makes no attempt to systematically explore the possible input space via a metric like NC but finds that GAN-based transformations could expose new faulty behaviors.

DeepGauge expands on the idea of NC [194] by introducing three new neuron-level coverage criteria and two layer-level coverage criteria to produce a multi-granular set of DNN coverage metrics. To argue for the utility of these metrics, DeepGauge uses standard adversarial attack techniques [100, 162, 234, 39] to generate test suites. It then compares the NC of the original test suite against that of the new, augmented test suite, boosted by the generated adversarial examples. By doing so, it finds some evidence that adding adversarial examples tends to increase NC in terms of most of the proposed criteria. In Section 3.5, we report our results that explicit effort to increase NC actually does not improve defect detection and is often harmful in terms of naturalness and output impartiality.

Recent on-going work [76, 177, 273] found preliminary evidence that the correlation between NC and DNN robustness is rather limited and that similar structural coverage metrics for DNNs could be misleading. Specifically, their test suites are generated using the standard adversarial attack methods, and their evaluation is limited to defect detection only. Our study scope is more *comprehensive*: we use automated, quantitative measures of naturalness and output impartiality in addition to defect detection and systematically investigate the trade-offs; we design a novel diversity promoting regularizer to extend existing adversarial attack algorithms; and we include both classification models and regression models (8 models in total), as opposed to classification models only.

While our evaluation focuses on *generating* test suites, others focus on selecting existing tests based on model uncertainty [195] or surprise adequacy (*i.e., significantly different and adversarial*) [142].

Finally, it is worth noting that our proposed output impartiality criteria discussed in Section 3.4.3 is different from the concept of fairness in machine learning [56]. Fairness in ML is concerned with the bias of an ML model with respect to sensitive attributes, such as gender or race. Along a similar vein, Themis, a software fairness testing tool by Galhotra *et al.* [86], automatically detects causal discrimination between input-output pairs for user-specified attributes. In sharp contrast with these notions of fairness, our output impartiality is a measure of the bias on how a *test suite* exercises diverse output behaviors in an ML model.

Adversarial Attacks. Recent studies show that DNNs are vulnerable to adversarial examples [292, 100], *i.e.*, by adding a very small, often visually imperceptible, perturbation to an input, a well-trained DNN may produce misclassifications. While adversarial attacks employ a variety of methods to induce erroneous behavior, their effectiveness is largely measured by the attack success rate of the perturbed inputs and its distortion from the original inputs. Most optimization-based adversarial attacks [199, 39] are based on ℓ_2 or ℓ_∞ norm-based perturbation. Some work [238, 296] has attempted to improve or side step the norm constraint with domain specific transformations. In our evaluation of neuron coverage, we use the standard attack methods with ℓ_∞ norm constraint, because these methods are efficient and can generate natural examples.

Adversarial attack algorithms offer both targeted and untargeted attacks for perturbing inputs to be predicted as some other class. Untargeted attacks aim to turn the prediction into any incorrect class, while targeted attacks aim to turn the prediction into a specific class. We use untargeted attacks to give them more freedom to perturb the input in whichever way NC maximization incentivizes.

3.3 Study Methods

This section describes the datasets, DNN models, and adversarial attack algorithms used for our empirical study and describes our diversity promoting regularizer to increase neuron coverage.

3.3.1 Datasets and DNNs

Table 3.1 summarizes architectural details of all the DNNs under test.

CIFAR10 [155] is a dataset containing 32x32x3 RGB pixel images representing ten mutually exclusive classes of naturally occurring entities that are suitable for IS and FID realism measurement. We use two well-known pre-trained DNNs: a 56-layer ResNet [112, 123] and a 121-layer DenseNet [121, 242], both of which achieve competitive performance on this dataset.

MNIST [169] is a large, well-studied dataset containing 28x28x1 gray-scale pixel images representing handwritten digits from 0 to 9. For this dataset, we consider two fully connected neural networks: FCNet5 with 5 hidden layers and FCNet10 with 10 hidden layers, and two convolutional neural networks: Conv1DNet and Conv2DNet. Both convolutional neural networks have 2 convolutional layers followed by 2 fully connected layers, but vary the primary convolutional layer type from 1D to 2D. All MNIST DNNs were trained for 10 epochs using an Adam optimizer [144].

The two realism metrics we employ—IS [268] and FID [115]—are tuned on the internal structures of natural images which generally have both foregrounds and backgrounds. Because such naturalism is not applicable to a digit recognition task, we exclude MNIST when studying the relationship between NC and naturalness.

Udacity Self-Driving Car [302] is a dataset containing $480 \times 640 \times 3$ RGB pixel images

extracted from video footage shot by a camera mounted to the front of a moving vehicle and the corresponding angle of the steering wheel ($\pm 25^\circ$) for each frame. We use two pre-trained DNNs: DAVE2 and DAVE2-Norminit (abbreviated DAVE2-N), used by DeepXplore [238] and originally from NVIDIA [26].

Table 3.1: DNN Architectural Details

DNNs	Dataset	LayerType	# Layers	# Neurons
FCNet5	MNIST	Fully Connected	5	478
FCNet10	MNIST	Fully Connected	10	3,206
Conv1DNet	MNIST	Conv1D	4	35,410
Conv2DNet	MNIST	Conv2D	4	15,230
ResNet56 [112]	CIFAR10	Conv2D	56	532,490
DenseNet121 [121]	CIFAR10	Conv2D	121	563,210
DAVE2 [26]	Driving	Conv2D	10	82,669
DAVE2-N [238] ‘	Driving	Conv2D	10	82,669

3.3.2 Measuring Neuron Coverage

Pei *et al.* [238] formally define neuron coverage by the following:

$$neuron_cov(T, \mathbf{x}, t) = \frac{|\{n | \forall \mathbf{x} \in T, out(n, \mathbf{x}) > t\}|}{|N|}$$

where $N = \{n_1, n_2, \dots\}$ represents all the neurons in the DNN; $T = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$ represents all test inputs (i.e., those to be perturbed); $out(n, \mathbf{x})$ is a function that returns the output value of neuron n for a given test input $\mathbf{x}$ *scaled* to be between 0 and 1 based on the minimum and maximum neuron activations for the layer; and t is the user-set threshold for determining whether a neuron is sufficiently activated.

Figure 3.1 depicts an example neural network with a single hidden layer. Each circular node corresponds to a neuron organized and color-coded by layer. The hidden layer neurons also contain their layer-wise scaled activations in parentheses for comparison against a chosen threshold t . If $t = 0$, then $NC_{t=0} = 3/5 = 0.60$, or if $t = 0.75$, then $NC_{t=0.75} = 1/5 = 0.20$. Selecting an appropriate threshold t was an open issue in early NC research. When measuring NC, we vary a

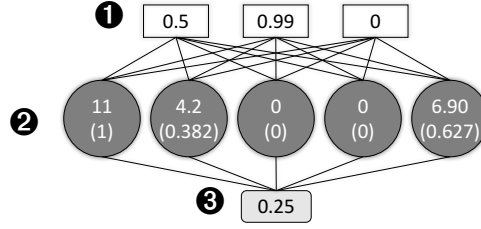


Figure 3.1: Single Layer DNN. ❶ represents inputs (i.e., pixels, features, etc.). ❷ represents a hidden layer of 5 neurons, where parentheses denote activations scaled between 0 and 1 for comparison against a NC threshold. ❸ represents an output layer of 1 neuron (i.e., class logits, probabilities, etc.).

threshold t for the range used by prior work [238, 296, 194], $t \in \{0, 0.2, 0.5, 0.75\}$.

3.3.3 Adversarial Attack Algorithms

Using adversarial attacks for test generation is analogous to fuzzing in software testing and acts as a means of introducing targeted perturbations. We select the following two adversarial attack algorithms [39, 199] due to their widespread usage in the ML literature.

Carlini-Wagner (CW) [39] constructs the adversarial example $\mathbf{x} + \boldsymbol{\delta}$, where $\mathbf{x}$ is the original input to attack, $\boldsymbol{\delta}$ is the adversarial perturbation, by solving the following optimization problem:

$$\min_{\boldsymbol{\delta}} \alpha \cdot L(h(\mathbf{x} + \boldsymbol{\delta}), y) + \|\boldsymbol{\delta}\|_p \quad \text{subject to} \quad \mathbf{x} + \boldsymbol{\delta} \in [0, 1]^n, \quad (3.1)$$

where y is the label of $\mathbf{x}$, L is a suitable loss function, h is the target model, $\|\cdot\|_p$ denotes the ℓ_p -norm such as ℓ_∞ , ℓ_0 , ℓ_2 norms, and α is a scaling constant to balance the the loss L and the ℓ_p -norm. The intuition behind the CW attack is to find some small perturbation $\boldsymbol{\delta}$ that we can add to the original input $\mathbf{x}$ such that it will lead the target model to change its classification. To achieve this, the CW attack exploits the loss function L to guide the generation of $\boldsymbol{\delta}$ that will make the target model's classification on $\mathbf{x} + \boldsymbol{\delta}$ different from $\mathbf{x}$. By minimizing the ℓ_p -norm of $\boldsymbol{\delta}$, the CW attack can ensure that such perturbation is small. In this effort, we use the ℓ_∞ norm, where distance is measured by the pixel with the greatest magnitude change from its original value. As for the loss function L , we use the loss function provided by [39] for our classification tasks. For our regression models, we substitute the standard CW loss function for a custom loss designed for

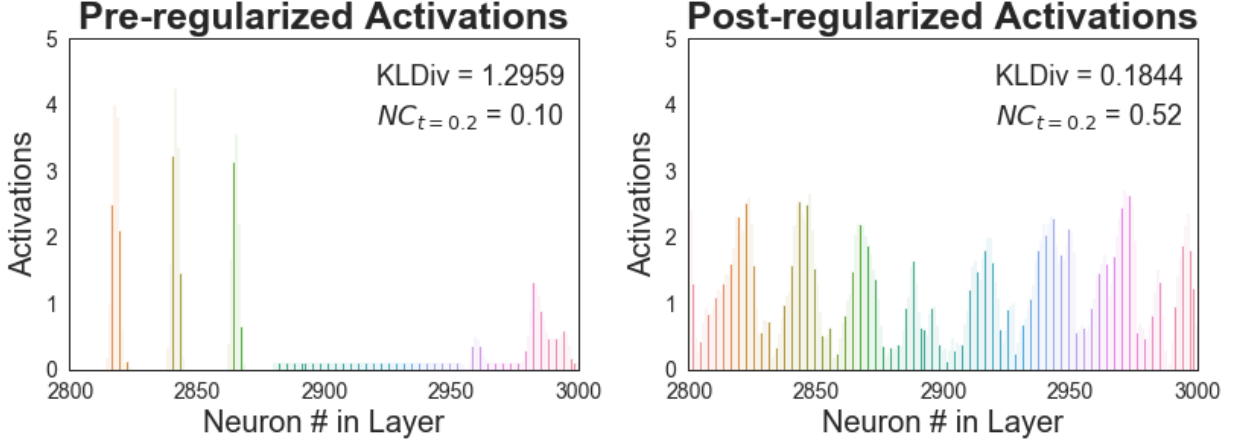


Figure 3.2: Neural activation before and after regularization: our regularization significantly promotes NC at $t = 0.2$.

regression tasks by Meng *et al.* [209].

Projected Gradient Descent (PGD) [199] finds the adversarial example $\mathbf{x} + \boldsymbol{\delta}$ by solving the following maximization problem:

$$\max_{\boldsymbol{\delta}} L(h(\mathbf{x} + \boldsymbol{\delta}), y) \quad \text{subject to} \quad \|\boldsymbol{\delta}\|_p \leq \epsilon,$$

where y is the label of $\mathbf{x}$, h represents the target model, L is the loss function for training h , ϵ is the perturbation limit. The maximization step will guide us to find the adversarial example and the ℓ_p norm constraint will make the perturbation small. For the PGD attack, projected gradient descent is performed to solve the above constrained optimization problem. We consider the ℓ_∞ norm constraint as in the CW attack, and use the sign of the gradient [100] to efficiently solve the maximization problem. For the loss function L , we choose the cross-entropy loss for classification tasks and mean square error for regression tasks. We vary a different perturbation limit $\epsilon \in \{0.1, 0.2, 0.3\}$ for the norm bounds to explore its possible effects on NC.

3.3.4 Extending Attacks to Increase NC

Adversarial attacks aim at creating perturbed inputs to achieve two primary objectives—maximizing loss while keeping ℓ_p -norm distance from the original inputs small. Previous research [238, 194] found that these algorithms do not produce any significant variation in NC. To increase NC while

Table 3.2: Original NC, Average % Increase from Original NC, and Maximum % Increase from Original NC

DNNs	NC _{t=0} (%)			NC _{t=0.2} (%)			NC _{t=0.5} (%)			NC _{t=0.75} (%)		
	Orig	Avg ↑	Max ↑	Orig	Avg ↑	Max ↑	Orig	Avg ↑	Max ↑	Orig	Avg ↑	Max ↑
FCNet5	96.09	0.56	0.66	61.21	22.59	61.72	15.01	45.75	173.98	4.33	38.72	171.88
FCNet10	78.52	7.65	18.63	16.79	54.68	98.90	3.70	38.27	71.63	0.89	63.77	123.91
Conv1DNet	68.08	4.82	14.50	12.67	8.77	45.33	1.26	12.85	139.58	0.48	15.16	63.38
Conv2DNet	94.96	1.77	4.18	23.89	9.11	36.47	6.66	25.48	55.69	1.23	67.88	122.04
ResNet56	95.07	0.22	0.40	26.87	4.31	11.77	5.42	6.17	21.76	1.29	6.20	15.71
DenseNet121	96.46	0.04	0.06	12.88	7.00	14.49	1.20	6.59	15.85	0.16	11.77	31.87
DAVE2	78.11	9.99	15.09	13.32	4.39	30.62	2.45	-5.78	9.28	0.72	-16.17	29.23
DAVE2-N	77.57	11.90	17.26	14.69	26.77	59.26	2.54	2.63	28.91	0.46	-1.26	37.14
Average	85.61	4.62	8.85	22.79	17.20	44.82	4.78	16.50	64.59	1.20	23.26	74.40

leveraging the skeleton of existing adversarial attacks, we design a novel adversarial attack regularizer to incorporate the maximization of NC as an additional objective. Our regularizer works by penalizing skewed layer-wise activations and thus promotes more diverse neural activation distributions. Diversity promotion has the effect gravitating all neurons toward the average magnitude of activation. Here we show the extended CW attack, augmented with our new diversity-promoting regularizer:

$$\min_{\delta} \alpha \cdot L(h(\mathbf{x} + \delta), y) + \|\delta\|_p + \lambda \cdot \sum_l \text{div}(\text{out}_l(\mathbf{x} + \delta), U)$$

subject to $\mathbf{x} + \delta \in [0, 1]^n$,

where $\lambda > 0$ is a user-set diversity weight to control how strongly we wish to induce higher NC; $\text{div}(\cdot)$ is a divergence function; $\text{out}_l(\cdot)$ is a function that returns the neural activations from the l^{th} layer of the DNN for the perturbed inputs $\mathbf{x} + \delta$; U represents a uniform distribution; and we consider ℓ_∞ norm in our method (i.e., choosing $p = \infty$). We use the Kullback-Leibler (KL) divergence [160] to implement our $\text{div}(\cdot)$ function, but any other measure of the distance between two probability distributions could be suitable. KL divergence measures how much information is lost by approximating the neural activations as if they were perfectly uniform—the higher the loss, the less diverse the activations. With a sufficiently high regularization weight placed on this objective, diversity promotion can induce previously inactive neurons to fire and increase NC. It is important to note that adding the regularizer does not necessarily harm the attack success rate

as approximately 23% of our generated suites have 100% attack success. However, there tends to be an inverse relationship between the regularization weight (λ) and the attack success rate. For example, the average attack success rate is 65% when λ is 0, and with increasing λ to 1, 10^1 , 10^2 , 10^3 , 10^4 , and 10^5 , the average attack success rate is 53%, 51%, 48%, 43%, 38%, and 35%, demonstrating some decrease. Figure 3.2 shows how our regularization promotes higher NC by having more neurons activated by visualizing neuron activation at a given layer in Conv2DNet.

Table 3.2 shows our regularizer’s effectiveness in terms of the average and maximum percent increases in NC over the baseline NC of the original test suite images for all models. Naturally, already highly activated DNNs are more difficult to activate further, making $NC_{t=0}$ undesirable for comparison purposes. On the other hand, $NC_{t=0.5}$ and $NC_{t=0.75}$ activate significantly smaller portions of the network. We report primarily on $NC_{t=0.2}$ for visual figures.

As an implementation note, our diversity-promoting regularizer can target a specific layer, contiguous and non-contiguous layer subsets, or all layers simultaneously. In our experiments, we vary the target layer one at a time, primarily to evaluate the sensitivity of NC to this regularization. For the MNIST models, we target each layer in turn. However, for larger models, we target k layers (default $k = 6$) evenly spaced in the model, starting from the first hidden layer and ending at the output layer.

Table 3.3: Experimental Variables

Variable	Values
Adversarial Attacks	CW, PGD
DNNs	FCNet5, FCNet10, Conv1DNet Conv2DNet, ResNet56, DenseNet121
Datasets	MNIST, CIFAR10
Target Layers	Varies
λ Diversity Weights	$0, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5$
c Confidence (CW) ¹	0, 20, 40
ϵ Limit (PGD)	0.1, 0.2, 0.3

³The parameter c encourages the solver to find an adversarial instance that is classified as a specific class with high confidence, see [39] for detail.

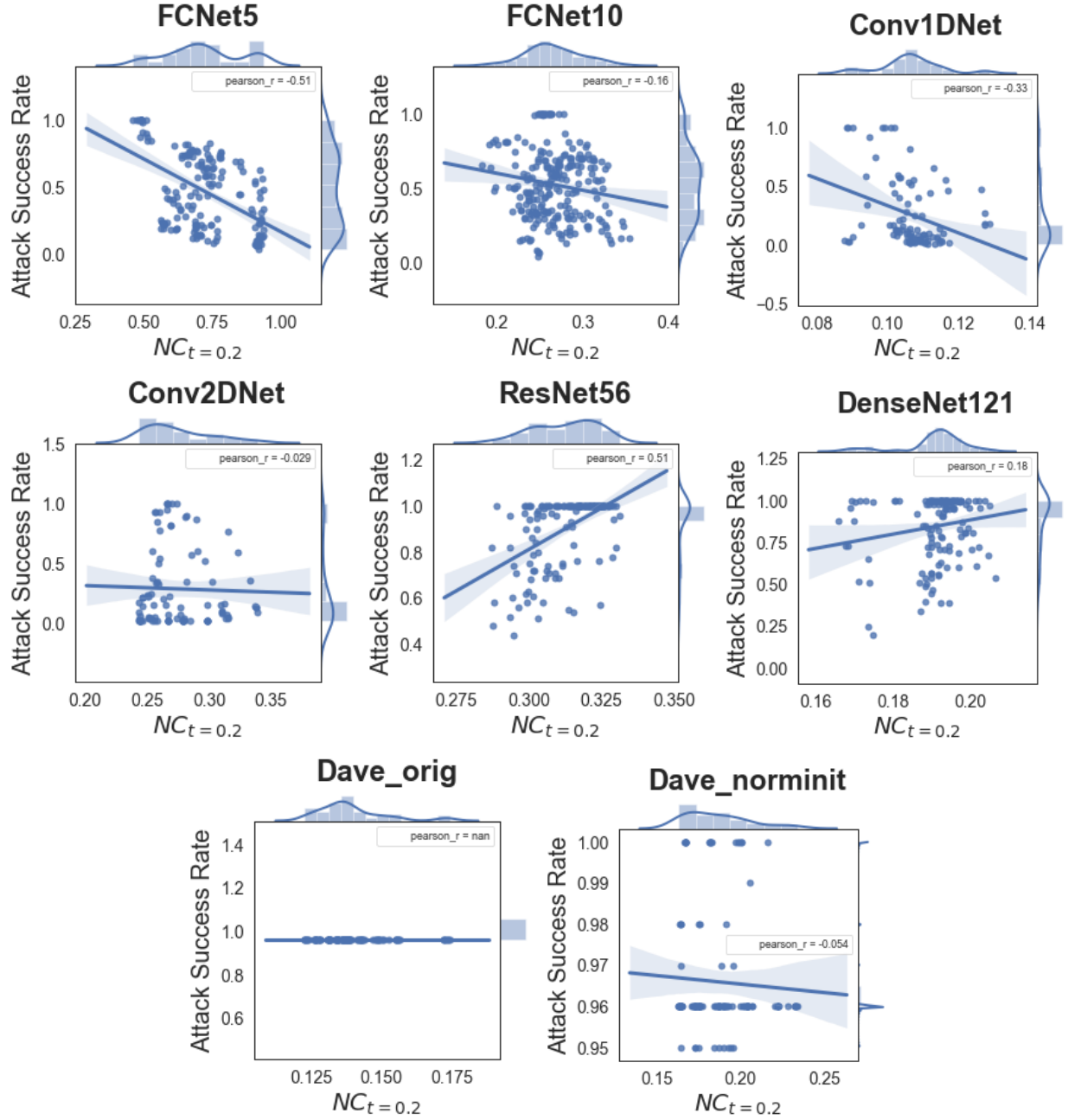


Figure 3.3: $NC_{t=0.2}$ vs ASR: the results show that NC does not consistently correlate with defect detection.

3.4 Findings

For each configuration, we construct a test suite of 100 randomly selected images such that each class is equally represented. This is to ensure that the suite has complete output impartiality before perturbation. We then use the NC-augmented adversarial attack algorithm to perturb the original tests before computing NC at threshold $t \in \{0, 0.2, 0.5, 0.75\}$, defect detection, IS, FID, and output impartiality. Finally, we perform an analysis of 2,095 test suites to measure the strength, direction, and significance of correlation. The experimental conditions are listed in Table 3.3.

Table 3.4: Correlation between NC & ASR: Gray indicates a p-value > 0.05

DNNs	CW - ASR Correlations				PGD - ASR Correlations			
	NC _{t=0}	NC _{t=0.2}	NC _{t=0.5}	NC _{t=0.75}	NC _{t=0}	NC _{t=0.2}	NC _{t=0.5}	NC _{t=0.75}
FCNet5	-0.20	-0.23	-0.18	0.07	-0.10	-0.52	-0.52	-0.32
FCNet10	-0.67	0.76	0.75	0.04	-0.18	-0.16	-0.10	0.14
Conv1DNet	NA	NA	NA	NA	0.58	-0.37	0.10	0.05
Conv2DNet	-0.16	-0.20	-0.29	-0.23	0.08	-0.04	-0.16	-0.36
ResNet56	-0.46	0.59	0.58	0.57	-0.11	0.52	0.53	0.21
DenseNet121	-0.83	-0.21	-0.06	0.13	0.19	0.18	0.20	0.11
Dave2	0.02	-0.17	-0.27	-0.21	0.30	-0.16	-0.45	-0.34
Dave2-N	NA	NA	NA	NA	0.00	-0.10	0.00	-0.08
Average	-0.38	0.09	0.09	0.06	0.10	-0.08	-0.05	-0.07

All correlations are presented in a tabular form and we visualize a sample of the NC_{t=0.2} results for PGD for presentation purposes. We adopt a standardized delineation of correlative significance laid out by Ratner [256] to characterize values between 0 and ± 0.3 as weak, ± 0.3 to ± 0.7 as moderate, and ± 0.7 to ± 1.0 as strong. Correlation coefficients are also color-coded according to whether or not they are statistically significant. Gray indicates a p-value > 0.05 and such values are discounted in our subsequent analysis. **Emboldened** values indicate that the results support the associated hypothesis and all others do not.

3.4.1 Defect Detection

3.4.1.1 Study Method

Since our approach relies on adversarial attacks to generate test suites, we equate the attack success rate (ASR) with defect detection rate (DDR) and use both measures interchangeably. Let $pert_acc$ represent the classification accuracy on the adversarially perturbed suite of test inputs (T), then DDR is simply $ASR(T) = 1 - pert_acc$. In order to use the same metric for the regression driving models, we discretize their continuous outputs into 25 equal-width intervals [298], each representing a 2° difference in steering angle.

3.4.1.2 Results

Figure 3.3 visualizes the relationship between NC and ASR, broken down by DNN for the PGD attack, which shows that NC is volatile and NC does not consistently correlate with defect detection. Even for models that share a large degree of architectural similarity, like the FCNet5 and FCNet10 models, the correlations differ in both strength and direction, reinforcing the unpredictability of NC.

Table 3.4 shows the results of all configurations broken down by an attack algorithm, network, and t threshold. Only 2 out of 64 correlations satisfy the hypothesis that NC is both positively and strongly correlated with defect detection. Independent of direction, 58% of experimental configurations show a weak correlation, while 25% are merely moderate. The correlation is positive in only 36% of configurations, negative in 52%, and non-existent in 12%.

Defect Detection. Our findings reject the hypothesis that NC is strongly and positively correlated with defect detection. Only 3% of the configurations supported this.

3.4.2 Naturalness

DL systems are designed to solve real-world problems and therefore a test suite must have realistic and natural inputs. In fact, several prior techniques are motivated by this naturalness goal and state

this requirement. For example, DeepXplore [238] uses domain-specific constraints to generate test images that are *valid and realistic*. DeepTest also states that it seeks to apply well-behaved transformations to preserve realism [344, 296]. We explicitly investigate whether maximizing NC can generate test suites reflecting the naturalness of the expected input space.

3.4.2.1 Study Method

Appraising the visual quality of an image can be highly subjective and there is still no definitive solution on how to formalize its naturalness. Fortunately, research into generative adversarial networks (GANs) [99] has produced several popular metrics for this purpose. We select the two most highly cited metrics from the GAN literature to objectively measure naturalness.

The **Inception Score (IS)** [268, 19] formalizes the concept of naturalness by decomposing it into the following two sub-concepts:

- *Saliency*. Of the possible class labels that could be applied to an individual image, only one has a high probability and the others are very low. This corresponds to the image being highly recognizable.
- *Diversity*. There are many different kinds of classes present across all images in the set.

The **Frèchet Inception Distance (FID)** [115, 216] is a measure of similarity between two datasets of images. It is calculated by computing the Frèchet distance between two Gaussians fitted to feature representations of the final average pooling layer within the InceptionV3 network [291]. The inventors, Heusel *et al.*, find evidence that FID captures the similarities of generated images better than IS and that FID correlates well with human judgement of visual quality. Unlike IS, the lower the FID value, the more realistic the images are, since the distance from the original images is smaller. Therefore, we investigate whether NC has a strong *negative* correlation with FID.

In the ML community, ImageNet [67] is considered as a comprehensive data set for image classifications. Thus, the authors of IS and FID derived these metrics based on the models trained on ImageNet and demonstrated generalizability to other datasets such as SVHN [219], CelebA [190],

Table 3.5: Correlation between NC & Naturalness: Gray indicates a p-value > 0.05

DNNs	CW - IS / FID Correlations				PGD - IS / FID Correlations			
	NC _{t=0}	NC _{t=0.2}	NC _{t=0.5}	NC _{t=0.75}	NC _{t=0}	NC _{t=0.2}	NC _{t=0.5}	NC _{t=0.75}
ResNet56	0.09 / 0.27	-0.87 / 0.76	-0.81 / 0.75	-0.59 / 0.59	0.34 / -0.03	-0.38 / 0.42	-0.52 / 0.46	0.06 / -0.14
DenseNet121	0.57 / -0.23	0.73 / 0.13	0.63 / 0.24	0.46 / 0.35	-0.15 / 0.26	0.16 / -0.06	-0.08 / 0.16	0.19 / -0.12
Dave2	-0.62 / 0.61	-0.32 / -0.22	0.02 / -0.31	0.21 / -0.18	-0.55 / 0.49	-0.89 / 0.97	-0.53 / 0.60	-0.29 / 0.29
Dave2-N	0.56 / 0.88	-0.48 / 0.50	-0.53 / 0.41	-0.50 / 0.49	-0.94 / 0.96	-0.67 / 0.78	-0.28 / 0.38	-0.23 / 0.32
Average	0.15 / 0.38	-0.23 / 0.29	-0.17 / 0.27	-0.10 / 0.31	-0.33 / 0.42	-0.45 / 0.53	-0.35 / 0.40	-0.07 / 0.08

CIFAR10 [155], and LSUN Bedrooms [325]. Therefore, we use the same method that the authors of FID and IS used. In our experiments, we exclude MNIST from the measurement of IS and FID, since it is inapplicable to discuss naturalness of highly, pre-processed MNIST digit recognition. Therefore, we use only CIFAR10 and driving datasets for examining the relationship between NC and naturalness.

3.4.2.2 Results

Figure 3.4 depicts the relationship between NC and IS and FID, broken down by metric, model for the PGD attack. Once again, the wide fluctuation of strongly negative and strongly positive correlations underscore the volatility of NC. Table 3.5 shows the results for each attack algorithm, model, and t threshold. Only 1 out of 64 correlations satisfy the hypothesis that NC is both positively and strongly correlated with improving input naturalness. Independent of direction, 38% of configurations show a weak correlation while another 45% are merely moderate. Independent of strength, the correlation is positive in only 31% of cases.

Unlike the mixed results for IS, increasing NC invariably increases FID, making the inputs less natural. In fact, **not a single** configuration in the FID experiment supports the hypothesis.

More than half of the PGD results across both IS and FID are statistically insignificant. This is because PGD attacks enforce a more strict ϵ perturbation limit, while the perturbations of CW attacks are theoretically unbounded and thus minimize the distortion as much as possible. Since this limit tightly constrains the range of measurements, it is difficult to assess the correlation with NC.

Figures 3.5 and 3.6 show a sample of two test suites with a 14% NC difference. While both sets

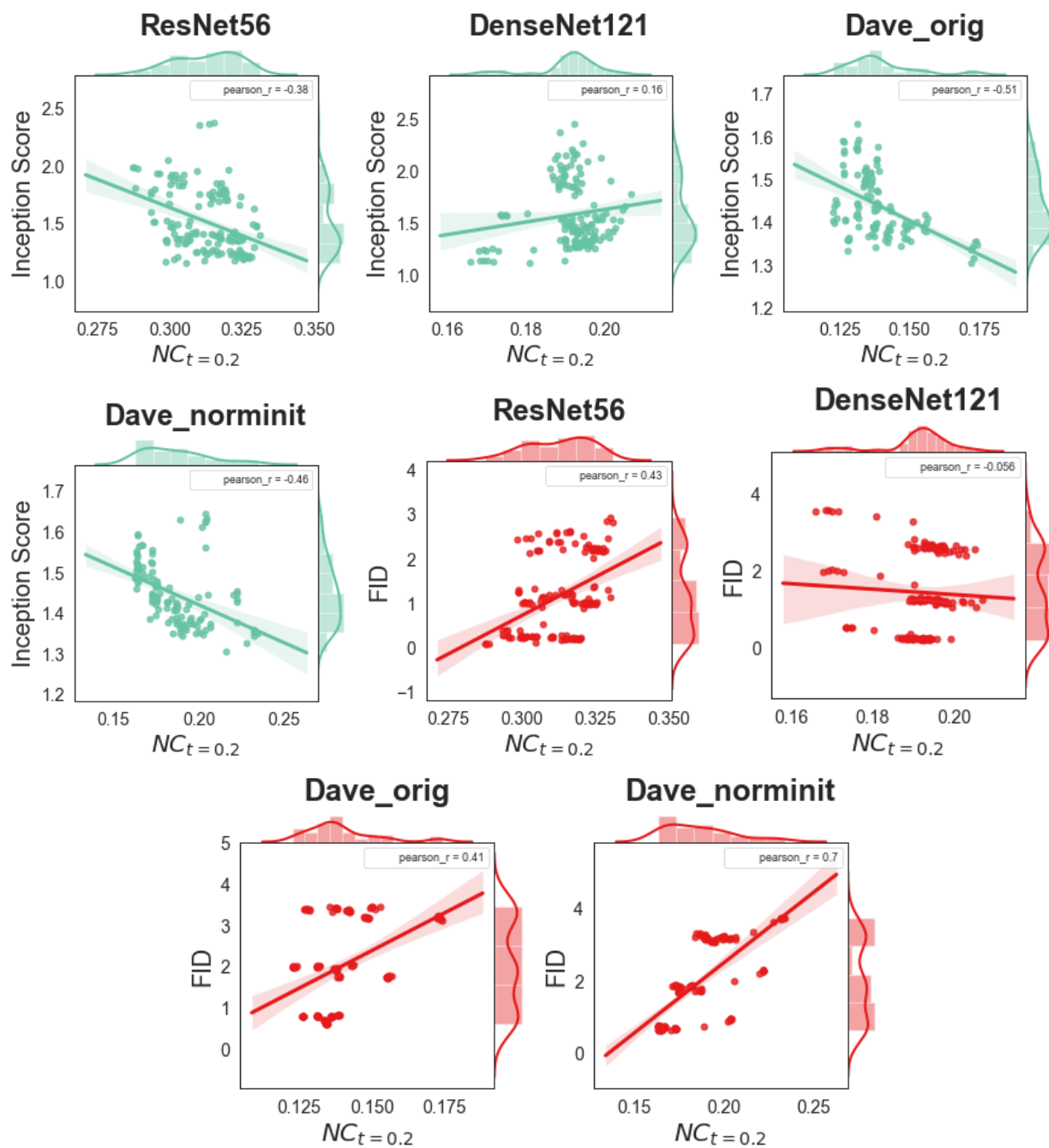


Figure 3.4: $NC_{t=0.2}$ vs Naturalness (IS / FID): the results show both strongly negative and strongly positive correlations.

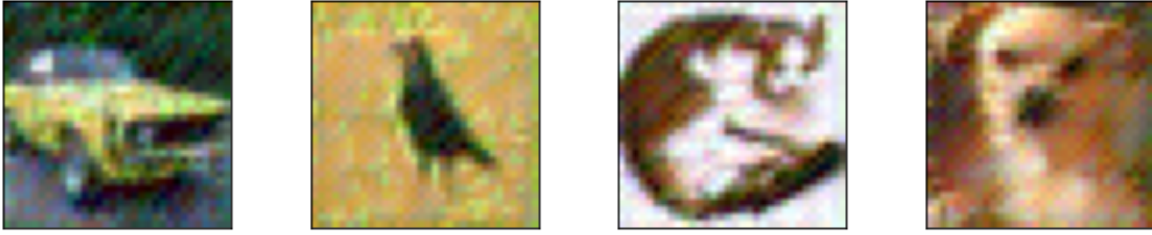


Figure 3.5: Test Suite #33. $NC_{t=0.2}$: 0.29 - IS: 1.97 - FID: 0.10

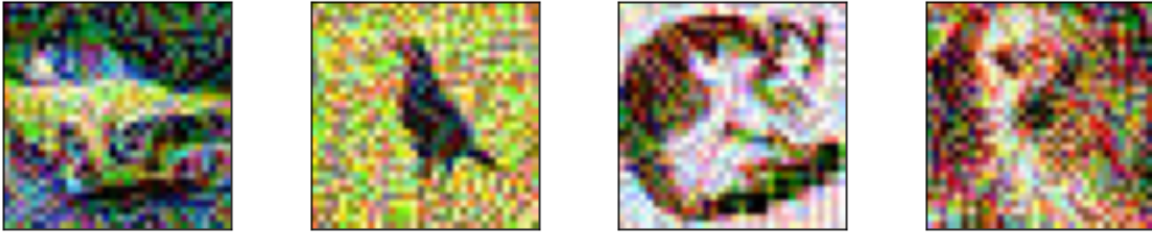


Figure 3.6: Test Suite #140. $NC_{t=0.2}$: 0.33 - IS: 1.48 - FID: 2.96

of images are noticeably distorted, test suite # 140 is clearly more unnatural. Test suite # 33 has an IS about 33% higher and an FID about 29x smaller, both confirming the intuition that Figure 3.5 with $NC = 0.29$ is more natural than Figure 3.6 with $NC = 0.33$. Here, increasing NC makes noisier and more noticeably perturbed inputs, thus a less valuable test suite.

Naturalness. Only 1.5% of all experimental results supported the hypothesis that NC is strongly and positively correlated with naturalness. 69% of the test suites are actually negatively correlated, implying that maximizing neuron coverage is likely to undermine naturalness.

3.4.3 Output Impartiality

The final dimension of our investigation probes the relationship between NC and the bias in model predictions. This idea of measuring the impartiality of model predictions is motivated by the output-uniqueness test selection criteria [2] in traditional software testing, which argues that a test suite must exercise diverse output behavior and should not prefer only a few output values. Investigating the relationship between NC and output impartiality is also motivated by several observations about

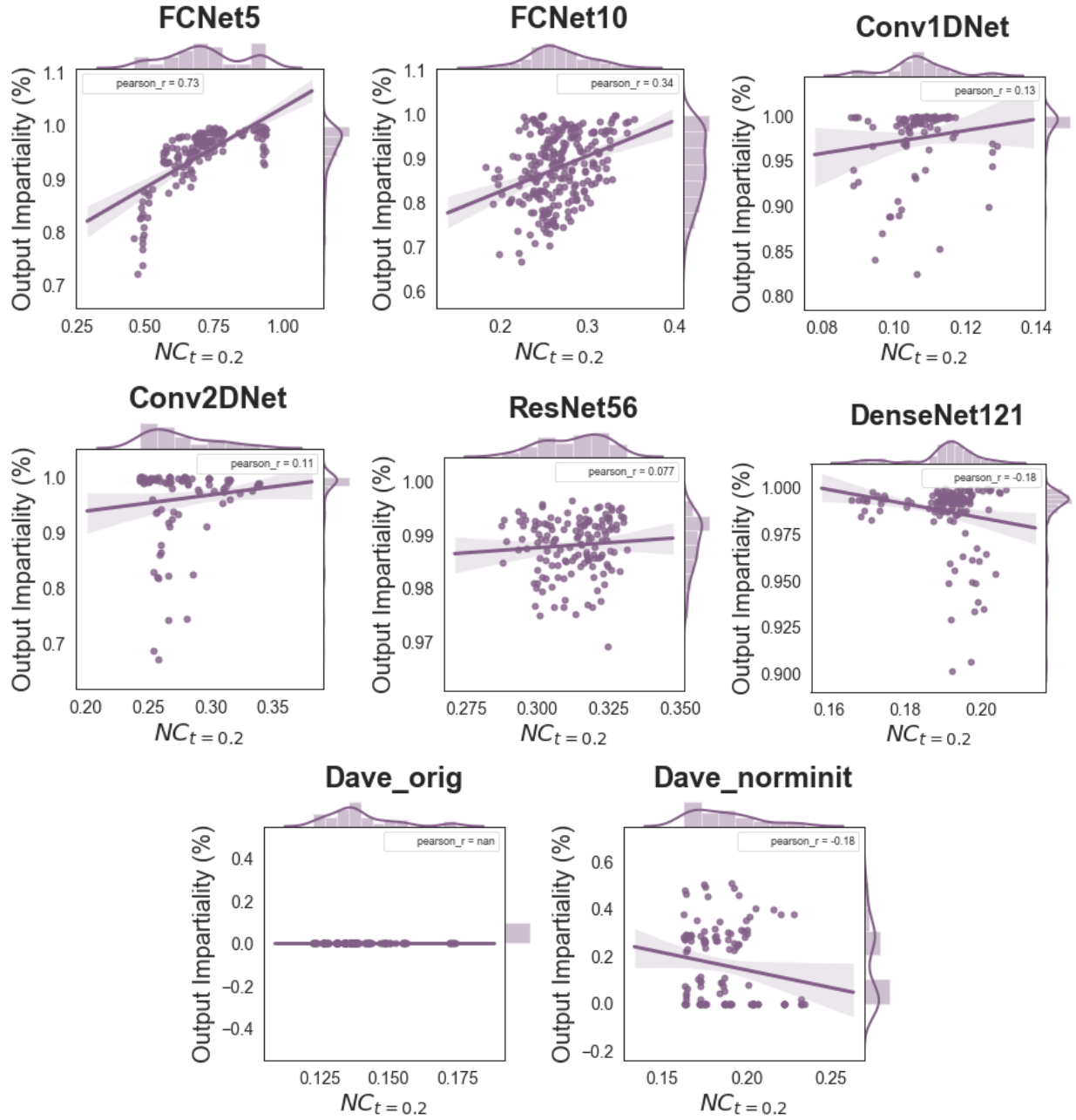


Figure 3.7: $NC_{t=0.2}$ vs Output Impartiality: the results show that increasing NC creates bias in output behavior.

Table 3.6: Correlation between NC & Output Impartiality: Gray indicates a p-value > 0.05

DNNs	CW - OI Correlations				PGD - OI Correlations			
	NC _{t=0}	NC _{t=0.2}	NC _{t=0.5}	NC _{t=0.75}	NC _{t=0}	NC _{t=0.2}	NC _{t=0.5}	NC _{t=0.75}
FCNet5	0.33	0.43	0.36	0.11	0.37	0.74	0.52	0.26
FCNet10	0.77	-0.70	-0.76	-0.02	0.42	0.34	0.25	0.02
Conv1DNet	0.39	0.08	0.10	0.15	-0.57	0.18	-0.28	-0.15
Conv2DNet	-0.22	-0.02	0.29	0.34	0.34	0.11	-0.04	-0.08
ResNet56	-0.27	0.45	0.43	0.43	-0.02	0.07	0.09	0.08
DenseNet121	0.79	0.11	-0.04	-0.18	-0.09	-0.18	-0.06	0.08
Dave2	NA	NA	NA	NA	-0.36	0.13	0.41	0.34
Dave2-N	0.66	0.01	-0.04	0.00	-0.05	0.19	0.26	0.22
Average	0.35	0.05	0.05	0.12	0.01	0.20	0.14	0.09

DNN behavior by prior work. Ilyas *et al.* [124] found that adversarial examples can be created by incorporating unnoticeable features of other classes to confuse the DL model. Similarly, Pei *et al.* found that different classes are associated with distinctive neuron activation patterns [238].

Consider a balanced test suite comprised of inputs evenly drawn from multiple classes. Suppose that the test suite is fed to a model and the model predicts always the same class label. This indicates output skew. Since one important aspect of testing is to exercise as much diverse output behavior as possible, we investigate the relationship between NC and the impartiality of predicted outcomes.

3.4.3.1 Study Method

We take inspiration for measuring impartiality from adjacent work on ecological biodiversity [185]. Instead of considering a distribution of species, we recast *impartiality* as a measure of the distribution of class predictions under a uniform input distribution (i.e. the initial test suite contains an equal number of inputs from each class). We use Pielou’s *evenness* score [243], a theoretically grounded measure of biodiversity [192] to assess the skew of the output class distribution. It uses a normalized Shannon’s entropy [274] scaled to a range of 0 and 1 by dividing the entropy of each test suite’s output distribution by the maximum entropy given the total number of classes. A high evenness score entails high impartiality (low bias). We define an output impartiality metric for a

test suite T with $|C|$ possible classes, indexed by k :

$$\text{output_impartiality}(T) = \frac{\sum_{t \in C_k} P_{t=C_k} \log P_{t=C_k}}{\log |C|},$$

where $|C|$ is the cardinality of classes and $P_{t=C_k}$ represents the percentage of the test cases t predicted to belong to class C_k . For the regression models, we use the same discretization method as before to enable the use of this metric.

3.4.3.2 Results.

Figure 3.7 visualizes the relationship between NC and output impartiality by DNN for CW. The results show that increasing NC creates bias in output behavior. Table 3.6 shows the results of all configurations by an attack algorithm and t threshold. Only 3 out of 64 configurations show that NC is both positively and strongly correlated with output impartiality. Independent of strength, the correlation is negative in 33% of correlations. Independent of direction, 62% of experimental configurations show a weak correlation while 32% are moderate.

3.4.3.3 Investigating Output Bias Caused by NC

In addition to the previous section’s correlation analysis, we design another experiment to investigate which classes are likely to be over-represented in the outputs after a test suite has been perturbed to maximize NC.

The idea of maximizing NC during test suite generation does not take into account that different classes of inputs can already have different baseline NC levels. For example, it may be the case that a set of inputs containing only the “dog” class in CIFAR10 has $\text{NC}_{t=0} = 0.9$ while another set of inputs containing the same number of “cars” has $\text{NC}_{t=0} = 0.6$. Increasing NC may then bias the perturbations—and therefore the output predictions—towards the class “car” with the higher NC baseline instead of “dog”. Below we describe an experiment conducted with the MNIST dataset to investigate this further.

We generate 10 partitions of the test data—one partition for each class—by randomly selecting

100 instances of that class from the test set. These partitions are then used to calculate a class-specific NC baseline. Since NC depends on the choice of t , we repeat NC baseline calculation for each class label, while varying t from 0 to 0.9 in an increment of 0.1. This process reveals which class label has the highest NC baseline, the second highest NC baseline, and so on. In other words, we rank class labels from the highest NC (Rank 1) to the lowest NC (Rank 10).

Suppose that class label 8 has a rank $\{1, 3, 3\}$ and class label 1 has a rank $\{8, 9, 10\}$, respectively for $t \in \{0, 0.5, 0.9\}$. A low average rank for class 8 (2.3) indicates that class 8 tends to have a high NC baseline regardless of t . On the other hand, a high average rank for class 1 (9) indicates that class 1 tends to have a low NC baseline. Therefore, during NC maximization, the perturbation process may favor over-representing class 8 in the output predictions. However, suppose that class 3 has an average ranking closer to 5—the midpoint of 10 possible labels. That implies that class 3 may have a high NC baseline under a certain threshold, but may have a low NC baseline under another threshold, or places the fifth for all t , etc. Thus, it would be unlikely for NC maximization to *consistently prefer* over-representation of outputs associated with class label 3 in the resulting test suite.

Table 3.7: MNIST Class and Average Rank of NC Baseline

Class	8	5	2	0	7	3	4	9	1	6
Average Rank	2.1	2.9	3.1	4.1	5.0	6.2	7.0	7.4	8.2	8.6

Concretely, an average rank closer to 1 indicates a greater likelihood of being over-represented in the output distribution through NC-maximization. Table 3.7 reports the average class ranks for the 10 class labels of MNIST. Here, we can see that class label 8 tends to have a high NC baseline and that class label 6 tends to have a low NC baseline across different thresholds. Therefore in the NC-maximized test suite, class 8 is likely to be over-represented and class 6 is likely to be under-represented in the output distribution.

We first construct a group of inputs with an output impartiality 1 by drawing 30 inputs per class label—every class is equally represented in the output distribution, because all are correctly predicted by the Conv2DNet trained for MNIST. We then use our test generation algorithm with a diversity-promoting regularizer to perturb the input set to increase its NC. The histogram on the left

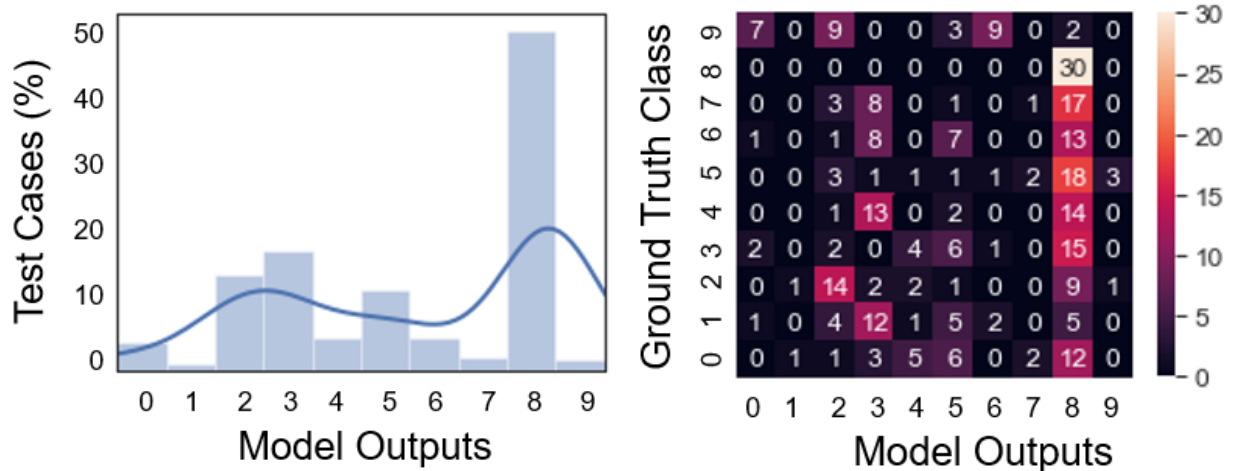


Figure 3.8: Output Prediction Distribution Histogram (left) and Cross-Class Prediction Heatmap (right)

in Figure 3.8 shows the percentage of model predictions for each class. The heatmap on the right details how many of the inputs belonging to each class in the original group were perturbed into predicting another class label. This perturbed test suite had a $NC_{t=0.2}$ of 0.34—about 40% higher than the original images, but 45% of all predictions are now for class 8, demonstrating output skew (a low impartiality score of 0.26). As expected, the classes with the low NC baselines (e.g., 6, 1, and 9) are among the most under-represented. This shows that, when NC maximization is used as a guidance criterion, a test generation technique can easily satisfy this criterion by simply perturbing inputs towards the class label with the highest NC baseline.

Output Impartiality. Only 5% of all experimental results support the hypothesis that NC is both strongly and positively correlated with output impartiality. When a few class labels have higher NC baselines than the other class labels, increasing NC biases the test suite to predominantly incorporate the features of this preferred subset.

3.5 Discussion

This section includes additional evidence and rationale that questions the meaningfulness of neuron coverage.

3.5.1 DeepXplore & DeepTest Comparison

It is certainly possible that another method may create a natural test suite with high NC. Therefore we perform similar analysis on the test suites generated by DeepXplore [238] and DeepTest [296] to see whether similar trade-offs exist. We utilize the authors’ publicly available implementations to generate tests for the MNIST and Driving datasets. For DeepXplore, **not a single** correlation is sufficiently strong enough to support the three hypotheses that NC is positively related with defect detection, naturalness, and output impartiality. For DeepTest, only one correlation for output impartiality at $NC_{t=0.5}$ is strongly positive. In fact, our investigation finds that many images generated by DeepXplore and DeepTest turn rich driving scenes into completely white images, yet retain their original labels. No human or program can predict a steering angle from such an unnatural input.

While the results from DeepXplore and DeepTest may have been sufficient to warrant skepticism about NC, our NC-maximizing approach is easily applicable and systematic. First, it can probe the behavior of a single model, while DeepXplore’s differential testing requires multiple models. As DNNs become large and costly to train [29], differential testing may become less practical. Second, by directly extending adversarial attacks that maximize defective behavior and minimize the norm-distance from their original sources, the generated test suite is orders of magnitude more natural. For instance, our test suites have average FID scores $458\times$ and $3,887\times$ higher than those created by DeepXplore and DeepTest respectively. While it is certainly possible that yet another method may create a natural test suite with high NC, our comprehensive experimentation of 2095 test configurations suggests that increasing NC is unlikely to correlate with defect detection, naturalness, and output impartiality. Triangulation between these approaches increases confidence about the external validity of our findings.

3.5.2 How Meaningful Is a Neuron?

The viability of NC as a DNN testing metric is underpinned by the idea that “each neuron independently extracts a specific input feature” [238] rather than collaborating with other neurons. However, recent research into DNN visualization techniques [335, 227, 221] has demonstrated

that this is not so—neuron independence and local feature extraction do not accurately characterize DNN behavior. Instead, the neurons in a layer interact with one another to pass information to subsequent layers and NC does not capture the richness of such neuron interactions. While the probability that a neuron distinctly encodes a specific feature increases the deeper it is situated in the DNN, many of the neurons represent an amalgam of very different abstract concepts, like the visualization of pixels leading to high activations of certain neurons in Figure 3.9 [227]. This observation raises serious doubts about whether neurons are even the right semantic units for understanding DNN behavior, further questioning the viability of NC as a meaningful test metric.

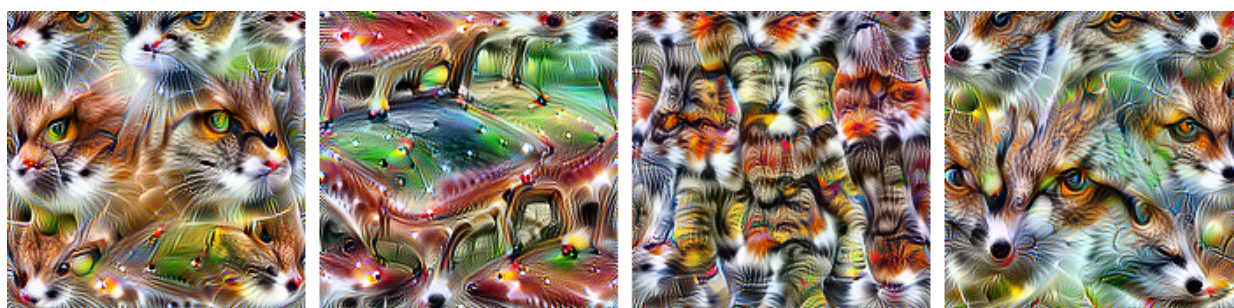


Figure 3.9: Visualization of neuron activations shows mixed concepts—cats, foxes, and cars [227]

3.5.3 Does NC Maximization Make Sense for Testing DNNs?

Assuming the best case scenario of neurons independently encoding specific features, is maximization of NC even desirable? Consider each neuron in a DNN as a binary classifier checking for the presence or absence of a specific feature within the input. For n neurons in a DNN, there are 2^n possible activation patterns. In general, establishing a single objective to maximize NC could easily target having *one* possible pattern, where all neurons are activated. As a simplified example, consider two neurons in a DNN trained for autonomous driving. Suppose that one detects the presence of vehicles and the other detects stop signs. NC maximization as a single objective in test generation can be easily satisfied with a single image containing both a vehicle and a stop sign together. Subsequently, such limited focus on NC could easily produce a test suite that does not cover other interesting portions of the potential input space.

3.6 Conclusion

Recent effort to test deep learning systems has produced an intuitive testing adequacy metric, called neuron coverage (NC) and its several variants. Prior work has also produced several test generation techniques that use NC as a guidance criterion and some has found evidence that adding adversarial inputs to an existing test suite tends to increase NC.

To systematically incorporate NC maximization to existing adversarial attack algorithms, we designed a novel diversity promoting regularizer that can be plugged into existing attack algorithms to increase NC. We then assessed the quality of the resulting test suites in terms of defect detection, naturalness, and output impartiality. From our evaluation of 2,095 experimental configurations involving 8 DNNs, 2 datasets, and 2 adversarial attack algorithms, we conclude that *NC should not be blindly trusted as a guidance metric for DNN testing*. While we do not claim that NC is useless, the main conclusion of this chapter answers **(RQ1)** in the negative and lend support to our hypothesis that structural coverage metrics generally harm important test objectives. In this particular study, increasing NC actually has a harmful effect by producing less natural inputs and by creating a skew in output distribution. This result is aligned with recent skepticism that code coverage in traditional software testing is not strongly correlated with test suite effectiveness and thus may not be a meaningful metric by itself [125]. We therefore advocate incorporating other test objectives such as *naturalness* and *output impartiality* and use multi-objective search techniques for testing DL systems.

Diversity objectives that characterize the output-space of a model (e.g. output impartiality) also subtly connect with other kinds of diversity to reveal potential testing issues. Firstly, observed skews in output diversity necessarily imply skew in the input features. In other words, the more the output distribution is skewed toward a particular class, the more those class features will be observed in the transformed test suite. Therefore, if a model is deterministic, increasing output diversity implies a commensurate increase to input diversity. Secondly, output diversity can be viewed as a proxy for structural coverage metrics as it provides some information on the model “state” in response to different inputs [226]. This suggests the possibility of bypassing the complexities of capturing structural coverages altogether and instead favor of the comparatively light-weight char-

acterizations of the data-space diversity instead. In the next work, we act upon the takeaways of this chapter and shift our attention to improving data-space diversity through data transformations.

CHAPTER 4

Sibylvariant Transformations for Robust Text Classification

Within software engineering and ML, data transformations are ubiquitously used to expand test and training sets and improve downstream performance metrics like model accuracy. The vast majority of these transformations employ small perturbations to preserve the underlying semantics of the original data and are therefore inherently limited in the amount of input coverage they can reach. Our hypothesis for **(RQ2)** is that one can drop the label-preserving constraint and jointly transform the input and label together to significantly expand data-space diversity. This expanded coverage was then evaluated against the downstream objectives of improving model generalization after training, and better defect detection and robustness during testing. These results were empirically studied primarily in the context of natural language, but the idea of sibylvariance can be used to diversify data in any domain (e.g. images, time series, tabular data, etc).

4.1 Introduction

Automatically generating new data is a critical component within modern machine learning pipelines. During training, data augmentation can expose models to a larger portion of potential input space, consistently leading to better generalization and performance [278, 156, 239]. After training, creating effective test instances from existing data can expose specific model failure modes and provide actionable corrective feedback [342, 262].

While many techniques can artificially expand labeled training sets or test suites, nearly all of them are class-preserving. That is to say, the model outputs are invariant (INV) with respect to the transformations. This cautious constraint ensures the new data does not lie in an out-of-distribution null class which might impede the learning objective. However, it also requires more conservative

transforms that inherently limit the degree of diversification.

In this work, we propose and extensively investigate the potential of *sibylvariant* (SIB) transformations that *knowably* vary the expected class. From the Greek *sibyls*, or oracles, the term parallels the oracle construction problem in software testing [18]. In a nutshell, sibylvariants either fully transmute a datum from one class c_i to another c_j , or mix data from multiple classes together to derive a new input with a soft label that reflects the mixed membership. In this way, SIB can more strongly perturb and diversify the underlying distribution. Moreover, SIB makes possible a new type of adaptive training by synthesizing data from frequently confused class pairs, challenging the model to differentiate with greater refinement.

In the following sections, we position SIB within a broader conceptual framework for all data transforms (Section 4.2) and highlight several newly proposed techniques (Section 4.3). To support a comprehensive evaluation of how SIB may complement or even surpass its INV counterparts, we implemented 41 new and existing techniques into an open source tool called `Sibyl`. Equipped with the framework and tool, we evaluate 3 central research questions:

- **RQ1. Generalization Performance.** Does training on SIB-augmented data improve model accuracy on the original test set?
- **RQ2. Defect Detection.** For models trained on the original dataset, how effective are SIB tests at inducing misclassifications?
- **RQ3. Adversarial Robustness.** Are models trained on SIB-augmented data more robust to existing adversarial attack algorithms?

Our comprehensive evaluation encompasses 6 text classification datasets, 11 transformation pipelines, and 3 different levels of data availability. In total, we trained 216 models and generated over 30 million new training inputs, 480,000 testing inputs, and 3,300 adversarial inputs. In the generalization study, SIB attained the highest accuracies in 89% (16 out of 18) of experimental configurations, with the adaptive mixture mutations being the most consistently effective. SIB also revealed the greatest number of model defects in 83% (5 out of 6) of the testing configurations. Lastly, of all the experimental configurations where adversarial robustness was improved over the no-transform

baseline, 92% (11 out of 12) of them involved SIB. Overall, our findings strongly support the efficacy of sibylvariance for generalization performance, defect detection, and adversarial robustness.

Lastly, we describe how SIB may operate theoretically and discuss potential threats to validity (Section 4.5) before contrasting it with related work (Section 4.6). The source code for `Sibyl` and our experiments is available at: <https://github.com/UCLA-SEAL/Sibyl>.

4.2 Sibylvariance

All data transformations in the classification setting can be categorized into one of two types:

- **Invariant (INV)** preserves existing labels.

$$\begin{aligned} \{T_{INV}(X_i), y_i\} &\rightarrow \{X_j, y_i\} \\ \text{where } X_i &\neq X_j \end{aligned} \tag{4.1}$$

For example, contracting “What is the matter?” to “What’s the matter?” should preserve a model behavior for sentiment analysis.

- **Sibylvariant (SIB)** changes an existing label in a knowable manner.

$$\begin{aligned} T_{SIB}(\{X_i, y_i\}) &\rightarrow \{X_j, y_j\} \\ \text{where } X_i &\neq X_j \text{ and } y_i \neq y_j. \end{aligned} \tag{4.2}$$

SIB transforms both the input X_i to X_j and the output label from y_i to y_j label, corresponding to the new X_j ; such transformation is analogous to mutating an input and setting a corresponding oracle in metamorphic testing [50]. For example, performing a verb-targeted antonym substitution on “I love pizza.” to generate “I hate pizza.” has the effect of negating the original semantics and will knowably affect the outcome of binary sentiment analysis.

It is important to note that transformation functions are not inherently INV nor SIB. The same exact transformation may have a different effect on expected model behavior depending on the

particular classification task. For example, random word insertions generally have an INV effect on topic classification tasks, but would be SIB with respect to grammaticality tasks [313].

4.2.1 Sibylvariant Subtypes

SIB can be further refined based on the types and degree of semantic shift in newly generated data:

- **Transmutation** changes one discrete kind into another, excluding the existing label, $L \setminus \{y_i\}$,

$$T_{SIB}(\{X_i, y_i\}) \rightarrow \{X_j, y_j\} \quad (4.3)$$

where $X_i \neq X_j$ and $y_j \in L \setminus \{y_i\}$.

Critically, the newly created data points retain stylistic and structural elements of the original that help boost diversity.

- **Mixture Mutation** mixes inputs from multiple classes and interpolates the expected behavior into a mixed label distribution (i.e. *soft label*). Equivalently, we have:

$$T_{SIB}(\{X_i, y_i\}) \rightarrow \{X_j, y_j\} \quad (4.4)$$

where $X_i \neq X_j$ and $y_j \in \bigcap_l^{|L|} \lambda_l$

where the final term indicates a λ -degree of membership in each label l belonging to the expected input space and is normalized as a probability distribution (i.e. $\sum_l \lambda_l = 1$). For example, a document with topic ‘surfing’ can be combined with another document with topic ‘machine learning’ to yield a new label with probability mass placed on both topics. While mixture mutations may seem unnatural, the intuition is that humans can recognize mixed examples and adjust their predictions accordingly. Models ought to do the same.

4.2.2 Adaptive Sibylvariant Training

One unique and promising aspect of SIB is to target specific class pairings dynamically during training. In much the same way that a human teacher might periodically assess a students’ understanding and alter their lesson plan accordingly, SIBYL computes a confusion matrix and constructs more examples containing classes for which the model has the most difficulty differentiating. For example, if a topic model most frequently misclassifies ‘science’ articles as ‘business,’ *adaptive* SIB (denoted as α SIB) will generate new blended examples of those classes in every mini-batch until the next evaluation cycle. At that point, if the model confuses ‘science’ for “health,” α SIB will construct new mixtures of those classes and so on. SIBYL supports built-in runtime monitoring for α SIB training.

4.3 Transformations

In SIBYL, we defined 18 new transforms and adapt 23 existing techniques from prior work [262, 214, 318] to expand the coverage of SIB and INV text transformations. At a high level, Table 4.1 shows these 41 transforms organized into 8 categories: *Mixture* (i.e., blending text), *Generative* (i.e. concept-based text generation), *Swap* (e.g., substituting antonyms, synonyms, hypernyms, etc.), *Negation* (e.g., adding or removing negation), *Punctuation* (e.g., adding or removing punctuation), *Text Insert* (e.g., adding negative, neutral, or positive phrases), *Typos* (e.g. adding various typos), and *Emojis* (e.g. adding or removing positive or negative emoji). We highlight several signature transforms here and provide a more detailed listing in Appendix 4.8.1.

Concept2Sentence (C2S). C2S is a two step process: (1) extract a short list of key concepts from a document and (2) generate a new sentence that retains critical semantic content of the original while varying its surface form, style, and even subject matter. To accomplish this, we leveraged integrated gradients [290, 244] to produce saliency attributions that identify the most relevant tokens for a given class label. We then generate a well-composed sentence from the extracted concepts using a pre-trained BART [172] model fine-tuned on the CommonGen dataset

Category	Transformations
Mixture	TextMix [†] , SentMix [†] , WordMix [†]
Generative	Concept2Sentence [†] , ConceptMix [†]
Swap	ChangeNumber, ChangeSynonym, ChangeAntonym, ChangeHyponym, ChangeHypernym, ChangeLocation, ChangeName, RandomSwap
Negation	AddNegation, RemoveNegation
Punctuation	ExpandContractions, ContractContractions
Text Insert	RandomInsertion, AddPositiveLink [†] , AddNegativeLink [†] , ImportLinkText [†] , InsertPositivePhrase, InsertNegativePhrase
Typos	RandomCharDel, RandomCharInsert, RandomCharSubst, RandomCharSwap, RandomSwapQwerty, WordDeletion, HomoglyphSwap
Emojis	Emojify [†] , AddEmoji [†] , AddPositiveEmoji [†] , AddNegativeEmoji [†] , AddNeutralEmoji [†] , Demojify [†] , RemoveEmoji [†] , RemovePositiveEmoji [†] , RemoveNegativeEmoji [†] , RemoveNeutralEmoji [†]

Table 4.1: Transformations currently available in SIBYL. New transforms that we defined are marked with [†].

[180].

Prior to generation, it is possible to apply other transformations to the extracted concepts to encourage diversity or knowingly alter the label. For example, on the left hand side of Figure 4.1 an antonym substitution produces a SIB effect by changing the extracted concepts from [‘stupid’, ‘worse’] to [‘intelligent’, ‘better’]. The new sentence exhibits a change in subject and style, but is correctly transmuted to have positive sentiment. C2S is thus an extremely promising transformation for diversifying text along both INV and SIB directions.

TextMix, SentMix, and WordMix. Mixture mutations, like mixup [340] and cutmix [338] from the image domain, take a batch of inputs and blend them together to form new inputs with an interpolated loss and they have shown robustness to adversarial attacks. TextMix translates this idea to the text domain by merging two inputs and interpolating a *soft label* according to the proportion of tokens belonging to the constituent classes. While TextMix does a straightforward concatenation, SentMix shuffles the sentences and thus encourages long-range comprehension. WordMix concatenates and shuffles all words, encouraging keyword-to-topic understanding when

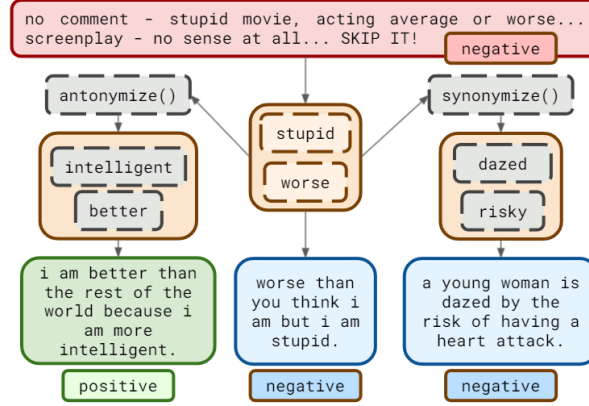


Figure 4.1: C2S intakes a text and its label (red) to extract keywords, ['stupid, worse']. These words are used to generate a new INV sentence shown in red. Alternatively, antonym (left) and synonym (right) substitution can produce new concepts that further boost diversity.

sentence structure is compromised.

4.4 Experiments

4.4.1 Transformation Pipelines & Datasets

To compare the potential of INV, SIB, and both (INVSIB) in aggregate, we construct a *transformation pipeline* (TP) [63, 324], where we uniformly sample n transformations of the selected kind to generate new $\{X_i, y_i\}$ pairs. We also create TP s that apply a single transform, T_{SINGLE} , to highlight the efficacy of C2S, TextMix, SentMix, WordMix and their adaptive versions, prefixed with α . In total, we evaluate 11 TP s per dataset, shown in Table 4.2.

Due to space limitations, we report the top performing TP of each kind using an asterisk (*). INV* represents the best from T_{INV} and T_{C2S} , while SIB* represents the best from T_{SIB} and the mixture mutations. For **RQ1**, we also compare against TMix [48], EDA [318], and AEDA [135]. TMix is a recent *hidden-space* mixture mutation for text, as opposed to SIBYL’s direct mixture mutation on the input space with greater transparency and examinability. EDA and AEDA are examples of recent INV transformations. Full results are available in the appendices.

We study six benchmarks for two kinds of NLP tasks: topic classification and sentiment analysis. Table 4.3 summarizes their relevant details. To simulate different levels of resource availability, we create three data subsets with by varying number of examples per class — 10, 200, and 2500.

Shorthand	Description
T_{ORIG}	0 transformations as baseline
T_{INV}	sample 2 INVs
T_{SIB}	sample 2 SIBs
T_{INVSIB}	sample 1 INV and 1 SIB
T_{SINGLE}	apply one from C2S, TextMix, SentMix, Word- Mix, α TextMix, α SentMix, α WordMix

Table 4.2: TP descriptions. TP s with an α -prefix use targeted, adaptive training (Section 4.2.2).

These subsets were expanded $30\times$ via augmentation for each TP . In total, we generated 144 new datasets ($144 = 6 \text{ benchmarks} * 3 \text{ levels of data availability} * 8 \text{ } TP\text{s which persist data. } \alpha\text{SIB is runtime only.}$)

Dataset	Source	Task	Subject	Classes	Test	Avg Len
AG News	[348]	Topic	News Articles	4	1,900	38
DBpedia	[348]	Topic	Wikipedia Articles	14	5,000	46
Yahoo! Answers	[348]	Topic	QA Posts	10	6,000	92
Amazon Polarity	[348]	Sentiment	Product Reviews	2	200,000	74
Yelp Polarity	[348]	Sentiment	Business Reviews	2	10,000	133
IMDB	[197]	Sentiment	Movies Reviews	2	12,500	234

Table 4.3: Dataset details. Test represents the number of examples per class in the test set.

4.4.2 Model Setting

We used a `bert-base-uncased` model [69] with average pooling of encoder output, followed by a dropout layer [287] with probability 0.1, and a single linear layer with hidden size 768 and GELU [114] activation. Maximum sentence length was set to 250. We use a batch size 16, an Adam optimizer [145] with a linear warmup, a 0.1 weight decay, and compute accuracy every 2,000 steps. All models were trained for 30 epochs on eight Nvidia RTX A6000 GPUs, with early stopping. In total, we constructed 198 different models.

For all TP s that produce a soft-label, we use a multi-class cross-entropy loss and computed

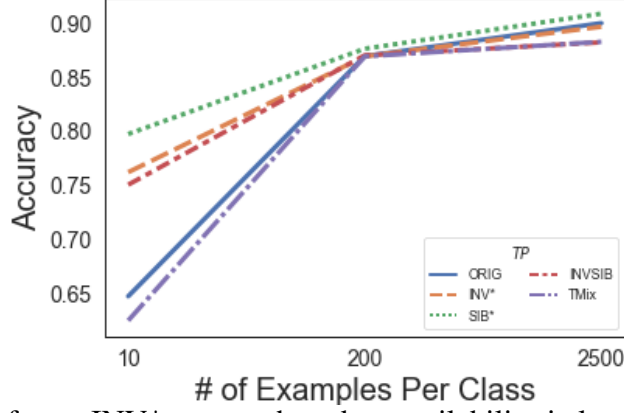


Figure 4.2: SIB* outperforms INV* most, when data availability is low, indicating the necessity of SIB to complement INV.

performance via a weighted top-k accuracy,

$$\sum_j^k \lambda_j \cdot \mathbb{1}(y_j = \hat{y}_j), \quad (4.5)$$

where λ_j is the degree of class membership, $\mathbb{1}(\cdot)$ is the indicator function, and y_j and $\hat{y}_j$ are the indices of the j -th largest predicted score for the ground truth label and predicted label, respectively.

4.4.3 RQ1. Generalization Performance

For RQ1, we explore how model accuracy on the original test set is influenced by training data augmented with INV and SIB transformations. Table 4.4 shows the results on six benchmarks with three levels of data availability.

We observe the most significant performance gains when training 10 examples per class —accuracy is improved by 4.7% on average across all datasets and by a maximum of up to 15% for IMDB. Figure 4.2 shows that as the number of labeled training data increases, a dominant trend emerged — T_{SIB} always generalized better to unseen test data. In fact, the only kind of transformation to always outperform both T_{ORG} and T_{Mix} is SIB*. Figure 4.3 shows the performance delta between INV* and SIB* against the T_{ORG} baseline at 200 examples per class. For every dataset, either $\alpha\text{SentMix}$ or $\alpha\text{TextMix}$ is the best performing TP , while INV* actually leads to performance decreases for DBPedia, Yahoo! Answers, and IMDB.

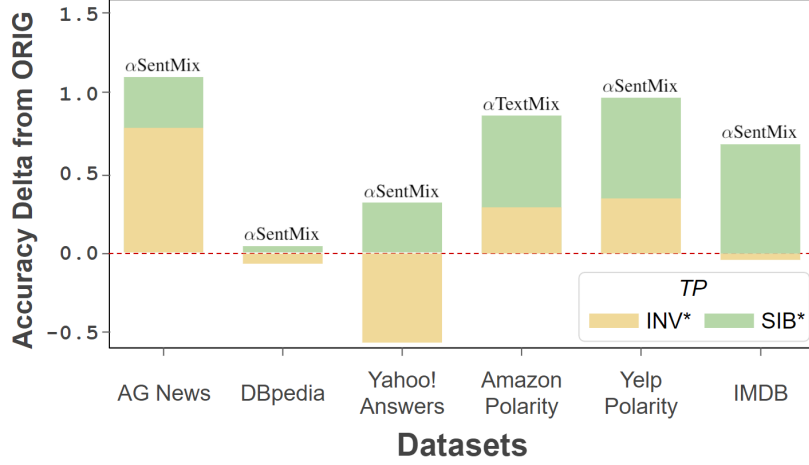


Figure 4.3: The best performing TP for each dataset trained on 200 examples per class. α SentMix or α TextMix leads to the highest performance gains. SIB* consistently outperforms INV*.

Dataset	TP	# Examples / Class			Dataset	TP	# Examples / Class			Dataset	TP	# Examples / Class		
		10	200	2500			10	200	2500			10	200	2500
AG News	ORIG	75.08	88.70	91.65	DBpedia	ORIG	95.71	98.87	98.96	Yahoo! Answers	ORIG	56.24	69.77	73.18
	INV*	84.28	89.46	91.95		INV*	97.29	98.81	99.00		INV*	61.39	69.21	72.53
	SIB*	83.52	89.80	92.42		SIB*	97.96	98.90	99.06		SIB*	62.47	70.10	73.37
	INVSIB	84.09	89.00	91.36		INVSIB	95.64	98.74	98.92		INVSIB	62.01	67.75	73.16
	TMix ‡	81.38	88.62	89.43		TMix ‡	97.51	98.66	98.89		TMix ‡	53.68	69.03	69.50
	EDA ‡	81.50	88.98	90.93		EDA ‡	97.42	98.63	98.89		EDA ‡	57.88	68.03	69.15
Amazon Polarity	AEDA ‡	81.03	88.74	92.09		AEDA ‡	97.30	98.88	98.89		AEDA ‡	59.51	67.37	69.91
	ORIG	67.30	89.22	92.08	Yelp Polarity	ORIG	74.62	91.66	93.70	IMDB	ORIG	64.70	86.96	90.02
	INV*	73.69	89.53	92.21		INV*	83.91	92.00	94.29		INV*	76.20	86.94	89.69
	SIB*	74.90	90.03	92.26		SIB*	80.46	92.60	94.69		SIB*	79.74	87.65	90.90
	INVSIB	73.50	89.06	91.26		INVSIB	78.90	91.85	93.03		INVSIB	75.04	87.04	88.24
	TMix ‡	62.14	87.98	91.00		TMix ‡	61.81	91.19	92.80		TMix ‡	62.45	86.94	88.29
	EDA ‡	59.40	87.68	92.20		EDA ‡	71.90	90.88	94.11		EDA ‡	67.37	86.45	89.07
	AEDA ‡	64.72	88.92	91.83		AEDA ‡	79.39	91.60	94.06		AEDA ‡	72.61	86.56	88.63

Table 4.4: RQ1 accuracy comparison for INV*, SIB*, and INVSIB against baselines ORIG, TMix [48], EDA [318], AEDA [135]. An asterisk (*) indicates the best performance observed across underlying TP s of each kind, while a ‡ indicates related works for comparison.

One key reason that aided SIB in attaining strong performance is the use of adaptive training. On average, crafting new examples that target the model’s primary confusions during training added approximately 1% to accuracy relative to mixing classes uniformly at random. This shows another unique benefit of sibylvariance that is not transferable to its INV counterparts.

While our full scale experiments show a clear trend that SIB generally outperforms INV, we primarily evaluated TP s combining multiple transforms instead of assessing the efficacy of each in isolation. Initially, this was a logistical decision due to computational limitations. To investigate each transformation’s effect individually, we conducted a small scale experiment training 756 models ((39 transformations + 3 α SIB) $\times$ 6 datasets $\times$ 3 runs) on 10 examples per class with a

$3\times$ augmentation multiplier. Based on this experiment, we then computed each transform’s performance by averaging the accuracy change relative to a T_{ORIG} baseline across all datasets. Table 4.5 shows the top ten best performing transforms, six of which employ SIB techniques. These results expand support for the overall conclusion that sibylvariance represents an especially effective class of transformations for improving generalization performance.

Transform	Type	Avg Δ (%)
α SentMix	SIB	+4.26
α TextMix	SIB	+3.55
RandomCharInsert	INV	+3.55
TextMix	SIB	+3.22
Concept2Sentence	INV	+2.70
AddPositiveLink	INV / SIB	+2.48
AddNegativeEmoji	INV / SIB	+2.45
SentMix	SIB	+2.33
ExpandContractions	INV	+2.15
RandomCharSubst	INV	+2.06

Table 4.5: Top ten individual transforms over a no-transform baseline averaged across all datasets. The INV / SIB types were SIB for the sentiment analysis datasets and INV for the topic classification datasets. See Table 4.11 in the Appendix for more details.

Generalization Performance.

Models trained upon SIB-augmented data attained the highest test set accuracy in 89% (16 out of 18) of experimental configurations, with the adaptive mixture mutations being the most consistently effective.

4.4.4 RQ2. Defect Detection

For RQ2, we assess how generating new tests with INV and SIB can expose defective model behavior. A single test is simply an $\{X_i, y_i\}$ pair and a test suite is a set of such tests. Defective behavior is misclassification, which is measured via a test suite’s accuracy. For each dataset D , we select a high-performing BERT model trained only on the original dataset without any augmentation. Then for each of eight TP s (excluding α SIB relevant to training only), we create 100 test suites, each containing 100 randomly sampled tests. This yields a total of 480,000 tests. We then report an average accuracy for each D and TP pair.

Figure 4.4 shows how defect detection is enabled by INV and SIB. With the exception of Yahoo! Answers, the models scored nearly perfect accuracy on T_{ORIG} ; however, when the same models are tested using data generated with INV and SIB, they struggle to generalize. Test data synthesized with SIB can reveal most defects in these models, indicating the value of *sibylvariance* in constructing test oracles for ML models in the absence of expensive human labeling and judgements.

Tests which lie outside the expected input distribution are not likely to be fair nor actionable. Since SIB transforms generally perturb data more aggressively than INV ones, they likewise possess more potential for creating unreasonable, out-of-domain tests of model quality. However, the positive results in **RQ1** may justify the use of SIB transformations as reasonable for testing. Had the newly transformed data truly belonged to a different distribution, model performance on the in-domain test set should have decreased as a result of dataset shift [247, 120]. In fact, we observed the opposite as model performance was consistently improved. This suggests that SIB transforms yield data that is tenably in-domain and therefore may complement INV transforms in exposing defective model behavior.

We theorize that the effectiveness of SIB-generated tests comes from the expanded objectives it permits. For example, T_{TextMix} assess whether the model can recognize which classes are present and to what degree. T_{SentMix} does the same but further scrutinizes long-range comprehension by broadly distributing related topic sentences. Datasets with lengthy inputs are particularly vulnerable to transformations of this kind. Lastly, T_{WordMix} forces the model to forgo reliance on text structure and evaluates keyword comprehension amidst noisy contexts. In contrast, most INV transformations involve minor changes — e.g. expand contractions — and test the aspect of language already well modeled from extensive pre-training. The INV C2S transform is an exception that drastically alters input and thus reveals more defects than other T_{INV} pipelines.

Defect Detection. Models tested with SIB-transformed data exhibited the greatest number of defects in 83% (5 out of 6) of experimental configurations.

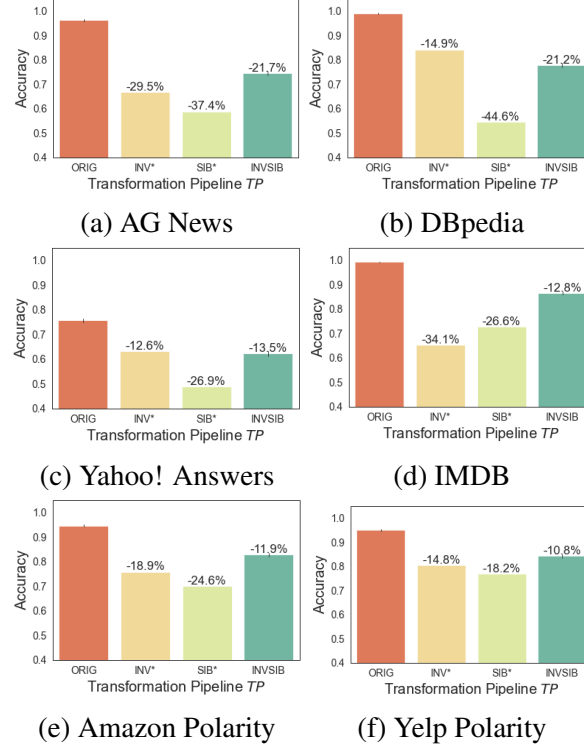


Figure 4.4: RQ2 defect detection comparison. Percentages show change in accuracy relative to T_{ORIG} . Lower accuracy indicates greater efficacy at inducing error.

4.4.5 RQ3. Adversarial Robustness

For RQ3, we assess whether models trained on INV or SIB are more resilient to adversarial attacks than models trained on an original data. An adversarial text input is typically obtained via semantic preserving (i.e. *invariant*) perturbations to legitimate examples in order to deteriorate the model performance. The changes are typically generated by ascending the gradient of the loss surface with respect to the original example and improving robustness to adversarial attacks is a necessary precondition for real-world NLP deployment.

We select three attack algorithms based on their popularity and effectiveness: (1) TextFooler [129], (2) DeepWordBug [87], and (3) TextBugger [175], all as implemented in TextAttack [214]. We focus on models trained with 10 examples per class because the largest changes in generalization performance are more likely to exhibit the clearest trend for adversarial robustness. For each of 11 models and 3 attacks, we randomly sample 100 inputs from the original data and perturb them to create a total of 3,300 adversarial examples.

Table 4.6 shows that, of all the cases where adversarial robustness is improved over T_{ORIG} , 92% of them involve SIB. On average, SIB*-trained models improve robustness by 4%, while INV*-trained models sustain a 1% decrease. Topic classification is made more robust via training with augmented data. Consistently, $T_{\alpha\text{-SentMix}}$ produces the most resilient models. For sentiment analysis, improved generalization performance enabled by SIB does not necessarily lead to improved robustness to existing adversarial attacks. The underlying sentiment models trained with augmented data improves generalization over T_{ORIG} by an average of 5%. However, counter-intuitively, the models are not more robust to the three attacks than T_{ORIG} and that Pearson correlation is -0.28 between accuracy and adversarial robustness. This finding motivates future work to investigate why there is a negative correlation and how to design SIB such that accuracy improvement also translates to corresponding adversarial robustness.

Dataset	TP	Attack Success Rate			Dataset	TP	Attack Success Rate			Dataset	TP	Attack Success Rate		
		TF	DWB	TB			TF	DWB	TB			TF	DWB	TB
AG News	ORIG	0.69	0.56	0.54	DBpedia	ORIG	0.92	0.55	0.64	Yahoo! Answers	ORIG	0.54	0.46	0.52
	INV*	0.66	0.56	0.48		INV*	0.76	0.47	0.48		INV*	0.57	0.49	0.49
	SIB*	0.60	0.43	0.45		SIB*	0.77	0.40	0.41		SIB*	0.48	0.41	0.49
	INVSIB	0.78	0.62	0.57		INVSIB	0.83	0.56	0.52		INVSIB	0.54	0.44	0.46
Amazon Polarity	ORIG	0.48	0.40	0.42	Yelp Polarity	ORIG	0.48	0.20	0.28	IMDB	ORIG	0.86	0.25	0.71
	INV*	0.49	0.42	0.36		INV	0.64	0.41	0.52		INV*	0.70	0.50	0.68
	SIB*	0.55	0.39	0.46		SIB	0.61	0.39	0.53		SIB*	0.56	0.32	0.55
	INVSIB	0.65	0.58	0.60		INVSIB	0.75	0.51	0.61		INVSIB	0.89	0.79	0.88

Table 4.6: RQ3 adversarial robustness comparison for INV*, SIB*, and INVSIB using TextFooler (TF), DeepWordBug (DWB), and TextBugger (TB). A lower attack success rate indicates a higher adversarial robustness.

Adversarial Robustness. Of all the experimental configurations where adversarial robustness was improved over the no-transform baseline, 92% (11 out of 12) of them involved models trained on SIB-augmented data.

4.5 Discussion

How does sibylvariance help? The primary purpose of data transformations in ML is to *diversify* datasets in the neighborhood of existing points, a principle formalized as Vicinal Risk Minimization (VRM) [44]. Synthetic examples can be drawn from a vicinal distribution to find similar but different points that enlarge the original data distribution. For instance, within image

classification, it is common to define the vicinity of an image as the set of its random crops, axial reflections, and other label-preserving INV transforms. While VRM can expose ML models to more diverse input space and consequently reduce generalization errors, the neighborhoods created by INV are relatively restricted. This is due to the label-preserving constraint limiting the degree of perturbation freedom on the original data.

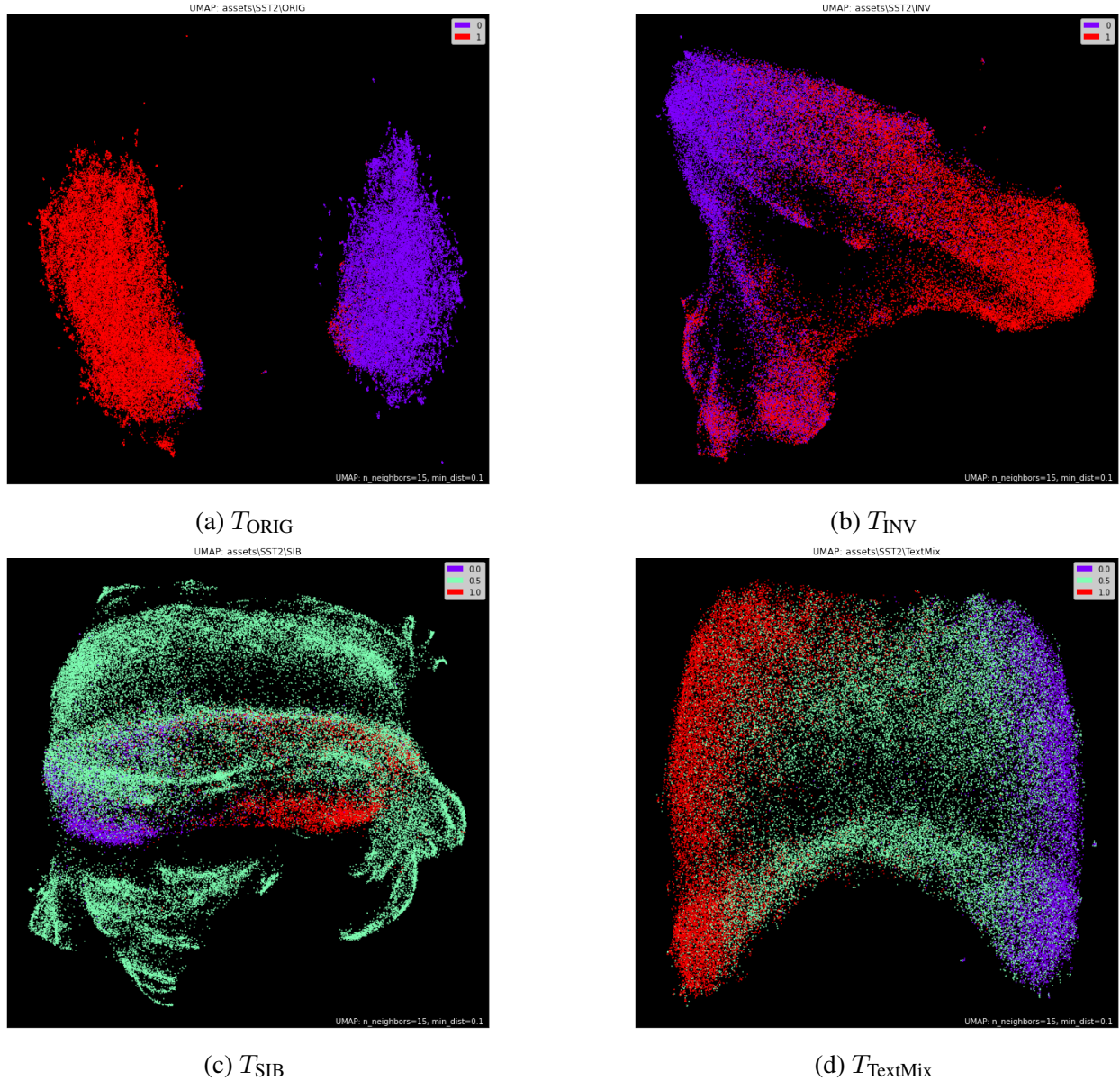


Figure 4.5: UMAP visualizations of BERT [CLS] tokens for SST-2. Blue, red, and green represent “Negative,” “Positive,” and “Mixed”, respectively.

SIB effectively expands the vicinity relation via transmutations and mixture mutations. Newly created data can claim full or mixed membership in target classes. To support our intuition, we

visualize the effects of various transformations on SST-2 [281]. Figure 4.5 presents the UMAP-reduced [205] [CLS] tokens produced by a BERT transformer for sentiment classification. Figure 4.5a shows that the classes are initially well separated and high performance can be obtained by selecting any separating surface between the two clusters. However, a more reasonable choice for the best boundary is one that exhibits the largest margin between classes — the very intuition behind Support Vector Machines [61]. Figure 4.5d suggests that a model trained on mixture mutations is likely to arrive at a boundary with the lowest loss. For example, in 4.5d, the augmented examples in green provide additional loss feedback from uncovered portions of the input space to encourage a decision boundary that maximizes the margin between class clusters. A similar expectation may hold for SIB in Figure 4.5c. However, the effects of INV transforms shown in Figure 4.5b do not appear to support such margin maximization.

Threats to Validity. External threats to validity include the generalization of our results to model architectures dissimilar to BERT (i.e. `bert-base-uncased`). It is possible that larger autoencoder models like RoBERTa [188] and auto-regressive models like XLNet [330] may respond differently to SIB transformations. Secondly, while the framework of sibylvariance is applicable to all data types, we have only provided empirical results supporting their efficacy for text classification models. We leave the exploration of SIB applications to image, time series, and other domains to future work.

Internal threats include how we derived mixed labels for generated text. We assumed that the critical semantics can be approximated via the ratio of words contributed by source text. This assumption may not account for other linguistic interaction and thus could lead to suboptimal labels. However, SIB did significantly improve upon the INV and the ORIG baselines in the **RQ1** generalization study, suggesting that the constructed soft labels still reflected useful semantics. This indirectly supports the validity of SIB-transformed data for testing in **RQ2**, although we acknowledge that additional caution is required for using any aggressively modified, synthetic data as a substitute for real data for the purpose of exposing defective model behavior.

4.6 Related Work

In this section, we broadly cover data transformations within and outside of the text domain because our proposed framework for sibylvariance is applicable to all classification contexts.

Data Augmentation. Effective data augmentation is a key factor enabling superior model performance on a wide range of tasks [156, 128, 324]. In many cases, practitioners leverage domain knowledge to reinforce critical invariances in the underlying data. In computer vision, for example, translation invariance is the idea that no matter where the objects of interest reside within an image, the model will still classify them correctly. Image translations and random crops encourage this more generalized conceptualization within the model [278] and all other transforms have a similar goal: reinforce a particular invariance that helps the learner perform well on future unseen data.

Numerous techniques have been proposed to assist with this learning objective and thereby improve generalization. Random erasing [350, 73] and noise injection [319, 324] support invariance to occlusions and promote robust features. Interpolating [27] and extrapolating [72] nearest neighbors in the input / feature space reinforces a linear relationship between the newly created data and the supervision signal while reducing class imbalance. However, nearly all of these approaches, and many others [276, 83], are label-preserving and therefore limited in their capacity to induce deeper learning of invariant concepts.

Sibylvariant transforms enjoy several desirable aspects of INV transformations while mitigating their drawbacks. Similar to feature space functions [72], mixture mutations do not require significant domain knowledge. Like approaches that reduce dataset imbalance [27], SIB transforms can increase class representation through mixed membership or targeted transmutations that inherit diverse characteristics of the source inputs. In all cases, relaxing the label-preserving constraint enables SIB functions to both complement and enhance the learning of critical invariances by further expanding the support of the dataset in new directions.

Adversarial Attacks & Robustness. Adversarial attacks are a special class of INV transformations that simultaneously minimize perturbations to the input while maximizing the perception of

change to a learner. This task is more difficult within the NLP domain due to the discrete nature of text, but several works [3, 346] have proven successful at inducing model errors. Real-world use of NLP requires resilience to such attacks and our work complements robust training [237] and robust certification [333, 245] to produce more reliable models.

Emerging Sibylvariant Transforms. Specific transformations designed to alter the expected class of an input have existed prior to this work [340, 338, 105, 354], albeit primarily in the image domain and also in a more isolated, ad hoc fashion. Among our primary contributions is to propose a unifying name, framework, and taxonomy for this family of sibylvariant functions. Furthermore, most prior works introduce a single transformation and evaluate its efficacy on training alone. In contrast, we proposed several novel transformations, a new adaptive training routine, and evaluated the broader impacts of 41 INV and SIB transforms on training, defect detection, and robustness simultaneously.

Recently published examples of SIB mixture mutations for text [106, 48] differ from ours in several important ways. Prior work operates exclusively within the *hidden* space inside specific models, which limits transferability between different algorithm types. All of our transformations operate in the *input* space, which is both more general and more challenging because we have to contend with rules of grammar and style. However, this also provides greater transparency. Furthermore, because our overall approach samples from 41 different transformations, we are able to exercise a broader range of model behaviors. For example, `SentMix` is designed to encourage long-range understanding, while other transforms evoke their own specific objectives. Any individual transformation is inherently more limited, e.g. `TMix` can only encourage the model to behave linearly for borderline cases.

4.7 Conclusion

Inspired by metamorphic testing, we proposed the notion of *sibylvariance* to jointly transform both input and output class (X_i, y_i) pairs in a knowable way. To explore the potential of sibylvariance, we define 18 new text transformations and adapt 23 existing transformations into an open source

tool called SIBYL. In particular, we define several types of mixture mutations and design a novel concept-based text transformation technique utilizing salience attribution and neural sentence generation. Across six benchmarks from two different NLP classification tasks, we systematically assess the effectiveness of INV and SIB for generalization performance, defect detection, and adversarial robustness. Our extensive evaluation shows that many SIB transforms, and especially the *adaptive* mixture mutations, are extremely effective. SIB achieves the highest training accuracy in 89% of the experimental configurations. When used for testing, SIB test suites reveal the greatest number of model defects in 5 out of 6 benchmarks. Finally, models trained on SIB-augmented data improve adversarial robustness $11\times$ more often than those trained on INV-augmented data. The results of this chapter provide clear and cogent evidence answering **RQ2** in the affirmative, that expanding data-space diversity through SIB transformations can dramatically improve the quality of NLP models in several important dimensions.

The primary difference between SIB and INV transforms is the degree of perturbation they introduce, which directly suggests the greater diversity-inducing power of sibylvariance is responsible for most, if not all, of the improvements. Unfortunately, this increased diversification may come at the increased risk that the labels derived for the new inputs do not align. The fact that generalization performance improves for SIB-trained models suggests that, at least *in aggregate*, our newly proposed SIB transforms generate reasonable labels. However, direct inspection of the data would be more helpful in understanding any specific failure cases where transformations are not helpful and should not be applied. To this end, we propose to evaluate the efficacy of data transformations in relation to the input features they affect. To support debugging functionality for undesirable transforms, we plan to leverage data provenance to identify effective transformations that can improve text augmentation.

4.8 Appendix

4.8.1 Implemented Sybil Transformations

Table 4.7: Transform descriptions currently implemented in `Sybil`, sampled from according to task (sentiment analysis or topic) and TP (INV, SIB, or INVSIB). Note that transformations are INV or SIB with respect to specific tasks. Asterisks (*) indicate that the variance type could be either INV or SIB, but the listed variance was judged to be more likely.

Category	Transformation	Sentiment	Topic
Mixture	TextMix	SIB	SIB
Mixture	SentMix	SIB	SIB
Mixture	WordMix	SIB	SIB
Generative	Concept2Sentence	INV	INV
Generative	ConceptMix	SIB	SIB
Word Swap	replace antonym	SIB	INV
Word Swap	replace cohyponym	INV	INV
Word Swap	replace hypernym	INV	INV
Word Swap	replace hyponym	INV	INV
Word Swap	replace synonym (wordnet)	INV	INV
Word Swap	change numbers (except 2 and 4)	INV*	INV
Word Swap	change locations based on dictionary	INV	INV
Word Swap	change names based on dictionary	INV	INV
Negation	add negation	INV*	INV
Negation	remove negation	INV*	INV
Punctuation	expand contractions	INV	INV
Punctuation	reduce contractions	INV	INV
Text Insertion	add URL to negative content	SIB	INV
Text Insertion	add URL to positive content	SIB	INV
Text Insertion	add negative phrase	SIB	INV
Text Insertion	add positive phrase	SIB	INV

(Continued on next page)

Table 4.7: Implemented `Sybil` transformations (continued)

Category	Transformation	Sentiment	Topic
Typos	char deletion	INV*	INV
Typos	char insertion	INV*	INV
Typos	char movement (n spaces)	INV*	INV
Typos	char repacement (homoglyph)	INV	INV
Typos	char replacement	INV*	INV
Typos	char swap (n spaces)	INV*	INV
Typos	char swap (QWERTY)	INV*	INV
Typos	word deletion	INV*	INV
Typos	word insertion	INV*	INV
Typos	word replacement	INV*	INV
Typos	word replacement (homophone)	INV	INV
Typos	word swap	INV*	INV
Emojis	replace words with emojis (Emojify)	INV	INV
Emojis	replace emojis with words (Demojify)	INV	INV
Emojis	add negative emoji	SIB	INV
Emojis	add neutral emoji	INV	INV
Emojis	add positive emoji	SIB	INV
Emojis	remove negative emoji	SIB	INV
Emojis	remove neutral emoji	INV	INV
Emojis	remove positive emoji	SIB	INV

4.8.2 Other Possible Text Transformations

Table 4.8: Transform NOT currently implemented in `Sybil`, but represent potentially interesting directions for future work. Asterisks (*) indicate that the variance type could be either INV or SIB, but the listed variance was judged to be more likely.

Category	Transformation	Sentiment	Topic
Word Swap	replace synonym (embedding)	INV	INV
Word Swap	word swap (masked)	INV*	INV
Word Swap	change gendered pronoun	INV	INV*
Word Swap	change protected class	INV	INV*
Word Swap	change "for" to 4	INV	INV
Word Swap	change "to" to 2	INV	INV
Word Swap	swap phrase with acronym	INV	INV
Negation	negation of negative clause	SIB	INV
Negation	negation of neutral clause	INV	INV
Negation	negation of positive clause	SIB	INV
Paraphrase	backtranslation	INV	INV
Punctuation	add exclamation	INV*	INV
Punctuation	add period	INV	INV
Punctuation	add question mark	INV	INV
Punctuation	remove exclamation	SIB*	INV
Punctuation	remove period	INV	INV
Punctuation	remove question mark	INV	INV
Text Insertion	add random URL (404)	INV	INV
Text Insertion	add neutral phrase	INV	INV
Tense / Voice	make continuous future tense	INV*	INV
Tense / Voice	make continuous past tense	INV*	INV
Tense / Voice	make continuous present tense	INV*	INV

(Continued on next page)

Table 4.8: Transformations not implemented (continued)

Category	Transformation	Sentiment	Topic
Tense / Voice	make perfect continuous future tense	INV*	INV
Tense / Voice	make perfect continuous past tense	INV*	INV
Tense / Voice	make perfect continuous present tense	INV*	INV
Tense / Voice	make perfect future tense	INV*	INV
Tense / Voice	make perfect past tense	INV*	INV
Tense / Voice	make perfect present tense	INV*	INV
Tense / Voice	make simple future tense	INV*	INV
Tense / Voice	make simple past tense	INV*	INV
Tense / Voice	make simple present tense	INV*	INV
Tense / Voice	change voice active	INV	INV
Tense / Voice	change voice passive	INV	INV
Emojis	replace emoji with word antonym	SIB	INV
Emojis	replace emoji with word synonym	INV	INV
Emojis	replace word with emoji antonym	SIB	INV
Emojis	replace word with emoji synonym	INV	INV

4.8.3 Sibylvariant Subtype Examples

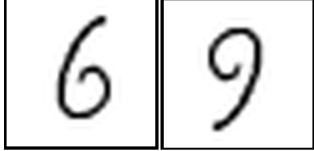
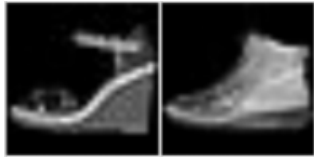
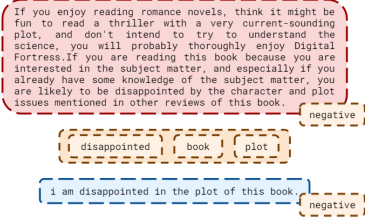
SIB Subtype	Image (Classification)	Text (Sentiment Analysis)
Transmutation $A \rightarrow B$ (Hard Label) Changes one class into another class, while retaining stylistic elements of the original.	Rotation  Digit 6 $\rightarrow$ Digit 9	Antonym Replacement I love NY $\downarrow$ I <u>hate</u> NY
	GAN-based Object Transfiguration  Sandal $\rightarrow$ Sneaker	Clause Negation You are a good person. $\downarrow$ You are <u>not</u> a good person.
		Stock Phrase Insertion It was a clever movie. $\downarrow$ It was a clever movie. <u>That said, I absolutely hated it.</u>
Mixture Mutation $A + B \rightarrow AB$ (Soft Label) Mixes two or more class labels into a single data point and then interpolates the expected behavior.	Mixup [340] Cutmix [338]  $[1, 0] + [0, 1] \rightarrow [0.35, 0.65]$	TextMix virutally unwatchable... + a vivid, thoughtful... = virutally unwatchable... a vivid, thoughtful... $[1, 0] + [0, 1] \rightarrow [0.17, 0.83]$
	Tile  $[1, 0, 0, 0] + [0, 0, 1, 0] + [0, 1, 0, 0] + [0, 0, 0, 1] \rightarrow [0.25, 0.25, 0.25, 0.25]$	WordMix it is essentially empty + this is a visually stunning... = love visually is is... $[1, 0] + [0, 1] \rightarrow [0.33, 0.67]$

Table 4.9: Examples of SIB transformations for the image and text domains. For mixture mutations, we show a soft label proportional to the pixel and word counts of their constituent parts.

4.8.4 Detailed Training Results for RQ1

Dataset	TP	10	200	2500	Dataset	TP	10	200	2500
AG News	ORIG	75.08	88.70	91.65	Amazon Polarity	ORIG	67.30	89.22	92.08
	INV	84.28	89.46	91.95		INV	71.09	89.53	92.21
	C2S	82.82	87.84	91.43		C2S	73.69	86.76	90.20
	SIB	83.52	89.20	91.55		SIB	69.23	87.00	91.45
	TextMix	83.53	89.17	91.58		TextMix	68.20	88.63	91.46
	SentMix	83.56	89.28	91.49		SentMix	71.22	88.85	91.28
	WordMix	82.61	88.59	90.42		WordMix	60.27	85.40	87.68
	α TextMix	81.53	89.51	92.20		α TextMix	74.90	90.03	92.26
	α SentMix	77.28	89.80	92.42		α SentMix	64.19	90.01	92.16
	α WordMix	83.13	89.46	91.91		α WordMix	64.21	89.09	91.98
	INVSIB	84.09	89.00	91.36		INVSIB	73.50	89.06	91.26
	TMix ‡	81.38	88.62	89.43		TMix ‡	62.14	87.98	91.00
	EDA ‡	81.50	88.98	90.93		EDA ‡	59.40	87.68	92.20
	AEDA ‡	81.03	88.74	92.09		AEDA ‡	64.72	88.92	91.83
DBpedia	ORIG	95.71	98.87	98.96	Yelp Polarity	ORIG	74.62	91.66	93.70
	INV	97.29	98.81	99.00		INV	77.92	92.00	94.29
	C2S	96.23	98.36	96.41		C2S	83.91	89.59	92.80
	SIB	95.26	98.73	97.60		SIB	78.67	91.89	93.69
	TextMix	97.96	98.88	97.86		TextMix	79.27	91.07	93.36
	SentMix	97.95	98.86	99.01		SentMix	80.46	91.96	93.62
	WordMix	97.03	97.89	98.59		WordMix	74.47	88.39	92.12
	α TextMix	97.72	98.87	99.04		α TextMix	77.72	91.73	94.50
	α SentMix	96.38	98.90	99.06		α SentMix	76.63	92.60	94.69
	α WordMix	97.01	98.90	98.90		α WordMix	78.30	91.50	93.67
	INVSIB	95.64	98.74	98.92		INVSIB	78.90	91.85	93.03
	TMix ‡	95.76	98.53	98.55		TMix ‡	61.81	91.19	92.80
	EDA ‡	97.42	98.63	98.89		EDA ‡	71.90	90.88	94.11
	AEDA ‡	97.30	98.88	98.89		AEDA ‡	79.39	91.60	94.06
Yahoo! Answers	ORIG	56.24	69.77	73.18	IMDB	ORIG	64.70	86.96	90.02
	INV	60.24	69.21	72.53		INV	76.20	86.94	89.69
	C2S	61.39	67.31	70.60		C2S	70.18	85.67	86.98
	SIB	61.30	68.45	73.18		SIB	73.51	86.38	88.71
	TextMix	62.47	68.72	72.08		TextMix	73.23	85.24	89.45
	SentMix	60.95	68.72	72.07		SentMix	76.75	85.55	89.10
	WordMix	59.98	67.66	72.96		WordMix	67.15	84.19	88.23
	α TextMix	60.26	69.89	73.15		α TextMix	74.09	87.52	90.60
	α SentMix	59.10	70.10	73.00		α SentMix	79.74	87.65	90.90
	α WordMix	60.74	69.99	73.37		α WordMix	73.01	86.92	87.85
	INVSIB	62.01	67.75	73.16		INVSIB	75.04	87.04	88.24
	TMix ‡	53.68	69.03	69.50		TMix ‡	62.45	86.94	88.29
	EDA ‡	57.88	68.03	69.15		EDA ‡	67.37	86.45	89.07
	AEDA ‡	59.51	67.37	69.91		AEDA ‡	72.61	86.56	88.63

Table 4.10: Performance (test set accuracy (%)) for all TP s. The results are averaged across three runs. Models are trained with either 10, 200, or 2500 examples per class. TP s are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively. White with a ‡ indicates related works for comparison. For TMix, EDA, and AEDA, we used the author’s open source code with their default / recommended configurations to transform the training datasets. However, we maintained the same model training hyperparameters as our other TP s to facilitate fair comparisons with our work.

Transform	Type	Avg Δ (%)
α SentMix	SIB	+4.26
α TextMix	SIB	+3.55
RandomCharInsert	INV	+3.55
TextMix	SIB	+3.22
Concept2Sentence	INV	+2.70
AddPositiveLink	INV / SIB	+2.48
AddNegativeEmoji	INV / SIB	+2.45
SentMix	SIB	+2.33
ExpandContractions	INV	+2.15
RandomCharSubst	INV	+2.06
AddNeutralEmoji	INV	+1.90
RandomInsertion	INV	+1.72
AddNegativeLink	INV / SIB	+1.64
α WordMix	SIB	+1.62
ChangeNumber	INV	+1.44
AddPositiveEmoji	INV / SIB	+1.25
InsertNegativePhrase	INV / SIB	+1.15
RemoveNegation	INV	+1.00
WordDeletion	INV	+0.86
RandomSwapQwerty	INV	+0.83
RandomCharSwap	INV	+0.77
ContractContractions	INV	+0.69
Emojify	INV	+0.59
ChangeLocation	INV	+0.37
Demojify	INV	+0.34
AddNegation	INV	+0.13
WordMix	SIB	+0.08
ConceptMix	SIB	-0.11
RandomCharDel	INV	-0.16
RemovePositiveEmoji	INV	-0.24
RandomSwap	INV	-0.28
ImportLinkText	INV	-0.56
ChangeHyponym	INV	-0.63
RemoveNeutralEmoji	INV	-0.72
RemoveNegativeEmoji	INV / SIB	-0.80
ChangeName	INV	-0.84
InsertPositivePhrase	INV / SIB	-0.95
ChangeSynonym	INV	-1.26
ChangeHypernym	INV	-1.78
ChangeAntonym	INV / SIB	-2.82
HomoglyphSwap	INV	-3.78

Table 4.11: Performance (test set accuracy (%)) for individual transforms over a no-transform baseline averaged across all datasets. The INV / SIB types were SIB for the sentiment analysis datasets and INV for the topic classification datasets.

4.8.5 Detailed Defect Detection Results for RQ2

Dataset	TP	Test Suite Accuracy	Dataset	TP	Test Suite Accuracy
AG News	ORIG	96.22	Amazon Polarity	ORIG	94.68
	INV	89.77		INV	86.91
	C2S	66.67		C2S	75.78
	SIB	74.77		SIB	80.99
	TextMix	59.97		TextMix	79.83
	SentMix	60.48		SentMix	79.83
	WordMix	58.82		WordMix	70.08
	INVSIB	74.50		INVSIB	82.78
DBpedia	ORIG	99.04	Yelp Polarity	ORIG	95.15
	INV	93.27		INV	89.76
	C2S	84.17		C2S	80.39
	SIB	71.67		SIB	82.76
	TextMix	54.42		TextMix	80.67
	SentMix	57.09		SentMix	81.09
	WordMix	57.48		WordMix	76.91
	INVSIB	77.79		INVSIB	84.32
Yahoo! Answers	ORIG	75.64	IMDB	ORIG	99.25
	INV	69.71		INV	90.01
	C2S	63.08		C2S	65.15
	SIB	58.87		SIB	84.48
	TextMix	48.77		TextMix	78.42
	SentMix	51.82		SentMix	79.45
	WordMix	53.58		WordMix	72.64
	INVSIB	62.17		INVSIB	86.42

Table 4.12: Test suite accuracy (%) by dataset and *TP*. Lower accuracy indicates higher defect detection potential. *TP*s are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively.

4.8.6 Detailed Robustness Results for RQ3

Dataset	TP	TF	DWB	TB	Dataset	TP	TF	DWB	TB
AG News	ORIG	0.69	0.56	0.54	Amazon Polarity	ORIG	0.48	0.40	0.42
	INV	0.73	0.59	0.48		INV	0.49	0.42	0.36
	C2S	0.66	0.56	0.48		C2S	0.51	0.49	0.50
	SIB	0.80	0.69	0.49		SIB	0.68	0.55	0.63
	TextMix	0.78	0.60	0.45		TextMix	0.56	0.41	0.46
	SentMix	0.70	0.57	0.61		SentMix	0.58	0.47	0.46
	WordMix	0.84	0.71	0.60		WordMix	0.74	0.69	0.73
	α TextMix	0.77	0.60	0.57		α TextMix	0.55	0.39	0.48
	α SentMix	0.60	0.43	0.46		α SentMix	0.56	0.49	0.53
	α WordMix	0.79	0.64	0.55		α WordMix	0.74	0.69	0.69
	INVSIB	0.78	0.62	0.57		INVSIB	0.65	0.58	0.60
DBpedia	ORIG	0.92	0.55	0.64	Yelp Polarity	ORIG	0.48	0.20	0.28
	INV	0.76	0.47	0.48		INV	0.64	0.41	0.52
	C2S	0.85	0.59	0.56		C2S	0.76	0.58	0.66
	SIB	0.80	0.58	0.64		SIB	0.68	0.53	0.65
	TextMix	0.85	0.48	0.41		TextMix	0.76	0.61	0.67
	SentMix	0.96	0.69	0.69		SentMix	0.70	0.52	0.60
	WordMix	0.91	0.64	0.76		WordMix	0.78	0.72	0.76
	α TextMix	0.82	0.51	0.53		α TextMix	0.61	0.39	0.53
	α SentMix	0.87	0.40	0.51		α SentMix	0.94	0.77	0.87
	α WordMix	0.83	0.55	0.49		α WordMix	0.62	0.49	0.56
	INVSIB	0.83	0.56	0.52		INVSIB	0.75	0.51	0.61
Yahoo! Answers	ORIG	0.54	0.46	0.52	IMDB	ORIG	0.86	0.25	0.71
	INV	0.57	0.49	0.49		INV	0.70	0.50	0.68
	C2S	0.58	0.53	0.54		C2S	0.93	0.59	0.89
	SIB	0.56	0.50	0.53		SIB	0.71	0.47	0.71
	TextMix	0.58	0.47	0.50		TextMix	0.85	0.32	0.73
	SentMix	0.72	0.64	0.72		SentMix	0.80	0.46	0.78
	WordMix	0.65	0.52	0.63		WordMix	0.84	0.74	0.84
	α TextMix	0.54	0.47	0.49		α TextMix	0.56	0.32	0.55
	α SentMix	0.48	0.41	0.48		α SentMix	0.95	0.91	0.96
	α WordMix	0.66	0.59	0.61		α WordMix	0.73	0.52	0.68
	INVSIB	0.54	0.44	0.46		INVSIB	0.89	0.79	0.88

Table 4.13: Attack success by dataset and TP for three adversarial algorithms: TextFooler (TF), DeepWordBug (DWB), and TextBugger (TB). Lower attack success indicates higher adversarial robustness. TP s are color coded by their variant type, where orange and light green are invariant and sibylvariant, respectively.

CHAPTER 5

FADA: Learning Effective Data Augmentation Policies with Text Quality Metrics and Semantic Features

Our investigation into sibylvariant transformations (Section 1.3.1) confirmed the hypothesis (RQ2) that deliberately shifting class labels during data augmentation can unlock significant gains in data-space coverage, leading to improved model performance across generalization, defect detection, and robustness benchmarks. While sibylvariance offers a potent mechanism for diversification by moving beyond label-preserving constraints, it implicitly raises questions about the broader landscape of data augmentation. Practitioners face a growing, complex space of possible transformations, and determining which specific techniques, applied in which ways, yield the optimal balance of diversity and quality for a given task remains a significant hurdle.

Therefore, while Sibyl established the potential of more aggressive diversification, the next logical step involves developing systematic methods for navigating the vast search space of augmentation strategies and controlling for unwanted side effects like quality degradation. The Feature-Aware Data Augmentation (FADA) framework, presented next, directly tackles this challenge (RQ3). FADA moves beyond evaluating augmentation strategies solely based on downstream task performance post hoc. Instead, it integrates text quality metrics and semantic feature analysis directly into the policy learning process, enabling a more interpretable and efficient optimization that seeks to maximize beneficial diversity while actively preserving semantic integrity and linguistic quality.

5.1 Introduction

The past few years witnessed rapid progress of data augmentation techniques (e.g. back translation, synonym substitutions), with hundreds of transforms proposed to diversify text data with a wide variety of linguistic features [74, 109]. This abundance of options presents a challenge for NLP practitioners seeking to identify which augmentations are most effective and can generalize to unseen task settings.

Since most existing evaluation methods depend on observing the post-train impact of augmentations, they do not scale to a such a large search space. In contrast, we propose to study the use of cheaply computed text quality metrics to guide the selection of effective augmentations. The first metric we consider is *label alignment*, i.e. the degree to which the assigned label reflects the semantics of the text, as measured via confident learning [223]. The second is *fluency*, which captures the flow and naturalness of a text represented by an inverted perplexity score. Finally, *grammaticality* is the degree to which the text is correctly structured and is measured by the number of grammatical errors reported by a grammar tool [167]. A motivating correlational study described in Section 5.2 demonstrates that these text quality metrics correlate with inference-time model performance and are therefore likely to be useful for judging the efficacy of text augmentations applied to training datasets.

We also observe that a transform’s efficacy cannot be fully characterized without also considering its interactions with the features present in the input. For example, the common `RandomDelete` transform is generally safe and useful, but poses a more significant risk of degrading data quality when applied to negated text. Within sentiment analysis, deleting a single negation feature (e.g. "I do *s.t.* not like stand-up.") can completely invert the meaning without updating the label appropriately. In our experience, models tested with $\langle \text{RandomDeletion}, \text{contains_negation} \rangle$ interactions suffered an 11% decrease in accuracy on average, about $5\times$ more than to texts without negation features. Therefore, the question of when to apply a transform effectively must consider the inherent features of the data.

Most existing automated data augmentation (AutoDA) frameworks produce *dataset-level* policies that don’t take into account all the relevant features of the input. However, they do gen-

erally produce easily understandable policies that can support data exploration and debugging. On the other hand, there are approaches that learn *sample-level* policies specific to each instance [222, 351]. These require significantly more computation and often produce policies that are uninterpretable because they’re implemented as neural networks. Fortunately, there is a middle ground that is feature-aware, interpretable, and efficient to compute.

In this paper, we propose FADA, a novel **feature-aware data augmentation** technique that efficiently learns a policy on when to augment by observing transform-feature interactions. The underlying text features are extracted from the text’s abstract meaning representation (AMR) graph [15] and stored into a binary feature vector that indicates whether a text contains high-level concepts such as *co-references*, *names*, *time*, and so on. This feature vector allows us to carefully target transformations and observe their impact on a set of cheaply computed text quality metrics.

To guide our evaluation of text quality and the benefits of feature-aware data augmentation, we formulated several research questions:

- RQ1.** How does text data augmentation affect dataset quality?
- RQ2.** Does feature-aware data augmentation improve model generalization?
- RQ3.** Which quality metrics, if any, are strongly and positively correlated with post-train model generalization?
- RQ4.** Can learned augmentation policies transfer to new datasets and task domains?

We also highlight that FADA possesses several key benefits:

- Feature-level policies are interpretable and facilitate a comprehensive understanding of the high-level interactions between your transforms and specific features.
- The policies enable targeted *instance-level* augmentations, selecting the optimal transforms for each text based on its features.
- Learning a policy only requires a few hours on a single GPU.
- Policies often transfer surprisingly well across datasets and task domains.

5.2 Motivating Text Quality as Predictive of Augmentation Effectiveness

Our approach to approximating augmentation effectiveness via text quality metrics requires us to first understand the complex relationship data quality has with model performance. To this end, we conducted a preliminary correlational study between label alignment, fluency, grammaticality, and *inference-time* F1 performance. For each dataset (5.4.1), we locate an open-source model from Huggingface [321] that has already been fine-tuned to perform well on the original dataset. Then for each baseline augmentation technique (5.4.2) we generate several transformed datasets to permit for sufficient variety in text quality. The average alignment, fluency and grammaticality scores are computed for both the augmented dataset and the original dataset so that we can compare the effect the different transforms had on text quality as relative ratio. This is done to ensure that transforms are only prioritized or penalized based on the changes they introduced, not any original issues with the source text. Finally, we extract the predictions of the open source models on the augmented text and compute their F1-scores against the ground truth labels.

The label alignment, fluency, grammaticality, and F1-score ratios for each dataset are then submitted to a Pearson correlation analysis. Figure 5.1 shows the observed correlations across all datasets. We adopt the standard delineation of correlative significance laid out by [255] to characterize values between 0 and ± 0.3 as weak, ± 0.3 to ± 0.7 as moderate, and ± 0.7 to ± 1.0 as strong. As can be seen, label alignment almost perfectly correlates with inference-time model performance, while fluency and grammaticality enjoy moderate correlations. The relatively low correlation between fluency and grammaticality also suggests that they indeed reflect different linguistic dimensions.

We use these results to motivate the design of a new augmentation strategy that explicitly incorporates text quality to the degree that they are predictive of model performance. Specifically, when aggregating the effect a transform has on text quality, we use linear importance weights $w_a = 0.7, w_f = 0.1, w_g = 0.2$ for label alignment, fluency, and grammaticality, respectively.

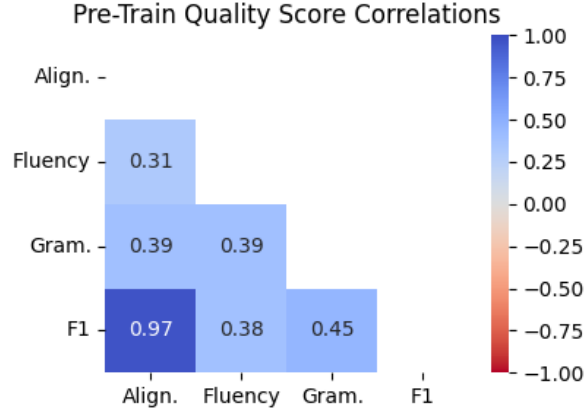


Figure 5.1: Quality metric correlations with F1-scores of already fine-tuned, open-source models.

5.3 Approach

In this section, we describe the design of FADA and its practical benefits. The principles outlined here may apply to any data modality, but we only focus on the text classification domain in our study.

5.3.1 FADA

The primary goal of FADA is to learn a feature- *and* instance-level augmentation policy, enjoying the benefits of nuanced transform targeting without the compute intensity of sample-aware policies. In this work, we formulate FADA as an automated, multi-objective search algorithm that learns an augmentation policy by observing the impact of transform-feature interactions quantified by a set of heuristic text quality metrics.

An overview of our approach is shown in Figure 5.2. As a preliminary step, we extract AMR graphs from each text in the train dataset D and then construct a binary feature matrix M_F using a set of featurizer functions that indicate which linguistic features (e.g. $f_1=\text{questions}$, $f_n=\text{co-references}$) are present in each text. The dataset D , binary feature matrix M_F , set of transforms T , and set of quality metrics Q are the primary ingredients for the policy search phase of the algorithm.

During the search phase, we iteratively sample interaction pairs that consist of one transform $t_i \in T$ and one feature $f_i \in F$. For each interaction, we use M_F to sample $n_{eval} = 128$ instances

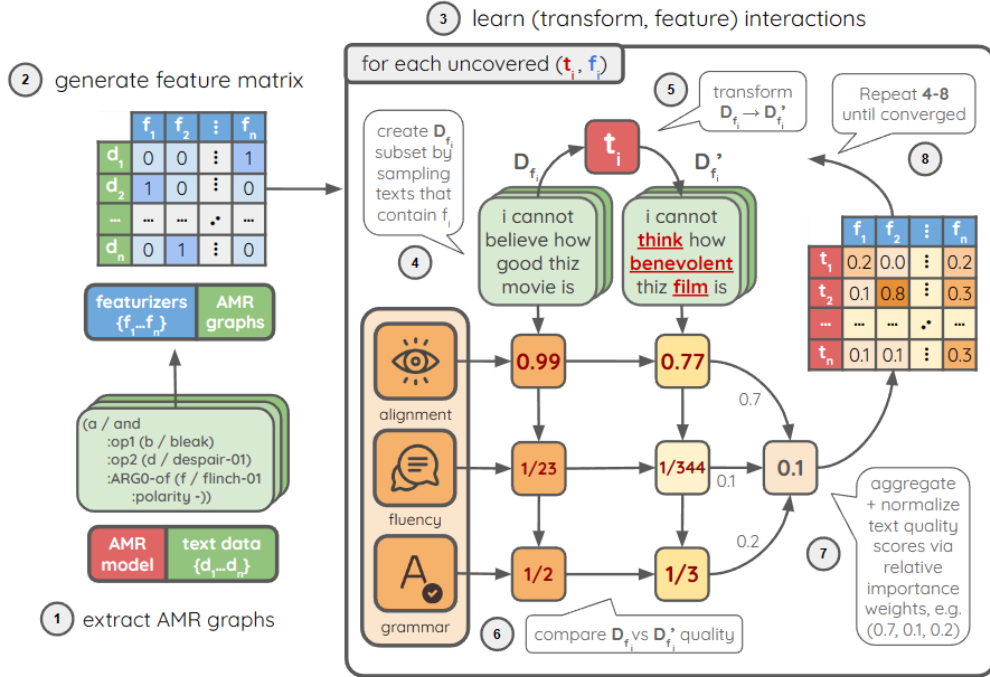


Figure 5.2: Overview of the FADA search procedure. In this sentiment analysis example, applying transform t_i to a data subset D_{f_i} containing feature f_i , results in adverse impacts to average alignment, fluency, and grammaticality. The transform-feature interaction is summarized into the augmentation policy with a score of 0.1, which indicates a relatively low probability of sampling t_i for any texts containing f_i .

from D into a data subset D_{f_i} containing feature f_i and transform it into D'_{f_i} by applying t_i . With these two subsets, we quickly compute and compare the text quality metrics to determine the effect t_i had on average alignment, fluency, and grammaticality. Here, a transform is judged according to the changes it introduced relative to the original state of the text. So if a text was initially disfluent and contained numerous grammatical errors, a transformation would only be penalized if it worsened those metrics. In order to reflect the relative importance of each metric, we linearly aggregate the quality scores using importance weights (See Section 5.2 for more details).

The aggregated quality score is then averaged into its corresponding (t_i, f_i) index of the policy matrix $\pi \in \mathbb{R}^{T \times F}$ and softmaxed to normalize into a proper probability distribution. Each element in π represents a conditional probability $P(t_i|f_i)$, indicating the suitability of a transform when applied in the presence of text bearing that feature. Even though this π is technically defined at the feature-level, it is possible to derive an instance-level policy π_{inst} by multiplying it into the binary feature matrix M_F and averaging the transform distributions if more than one feature is present. In this way, every instance in the dataset can be automatically and accurately targeted with augmentations that jointly maximize a weighted aggregate of the quality criteria.

The search phase iterates over additional rounds of sampling until the the policy converges to a stable state. Let π_1 and π_2 be policy matrices at different time steps, $|T|$ be the number of transforms, and $|F|$ be the number of features. We define convergence proportional to the search space via,

$$\|\pi_1 - \pi_2\|_2 < \frac{1}{|T| + |F|} \quad (5.1)$$

and in our studies, this threshold is $1/(20 + 44)$. Until convergence, FADA re-samples interaction pairs to control for variance and increase the likelihood of observing successful augmentations. Unsuccessful transformations generally occur because they apply to features that are not present in the text. For example, CHECKLIST’s `ExpandContractions` transform [261] has no impact on texts without contractions. To account for the *applicability* of a transform, we also adjust the probability of selecting a transform proportional to the rate at which it fails to make any change to the input text.

5.3.2 Text Transforms

A text transform is simply a function that intakes texts and alters them in some way. [258] distinguish two primary types of data-space transforms: (1) edit-based methods [316, 324] that employ simple, label-invariant sentence editing operations, such as word swaps, deletes, and typo insertion; (2) generation-based methods [151, 5, 248, 334] that utilize conditional generation models capable of synthesizing full paraphrases and other larger-scale alterations. Generation-based methods are generally better at preserving fluency and grammaticality, though at a much higher computational cost. In this work, we focus on the more common edit-based transforms because they are cheap to compute (i.e. enable more efficient search space exploration) and are more vulnerable to the kinds of disfluencies and grammatical errors we monitor. A detailed list of the 20 edit-based transforms used in this work is available in Appendix 5.8.2.

5.3.3 Text Features

The particular features explored by our approach are expected to have a large impact on the quality of the learned policies. While there exist many measurable properties of text, we decided to derive our features from the nodes, edges and attributes of AMR graphs [15] for several reasons. First, AMR offers a high-level, easily interpretable ontology of semantic relations embedded in text with clear correlates to the features explicitly targeted by different transforms. For example, `ChangeName`, `ChangeLocation`, `ChangeNumber` [109] all cleanly map to one some AMR relation type, e.g. `:name`, `:location`, `:quant`. Second, while AMR is primarily concerned with semantic features, it also provides coverage for some syntactical features, e.g. `amr-unknown` indicates the presence of a question mark, which is relevant to the `YesNoQuestionPerturbation` [74]. Third, semantic features are sparse enough to permit sufficient differentiation between texts, which in turn allows us to target augmentations with greater nuance. Finally, extracting accurate AMR graphs is conveniently available through open source tooling and we use `amrlib` [24] to annotate our studied datasets.

5.3.4 Quality Metrics

In contrast to previous approaches that rely on compute intensive model training to approximate augmentation effectiveness, we directly evaluate the impact of a transformation on different dimensions of its output text. The metrics chosen to quantify these dimensions — alignment, fluency, and grammaticality — reflect the intuition that classification performance is connected to intelligible, natural, and well-structured inputs. Next we briefly describe each of the quality metrics we studied.

Label Alignment. Label Alignment refers to the degree to which the label reflects the semantics of the text. We measure alignment using `cleanlab` [60], which implements a technique called confident learning (CL) [223]. CL identifies and characterizes label errors in datasets by estimating the joint distribution between noisy labels and trusted labels. Here, a surrogate model fine-tuned for the target dataset is queried to provide confidence estimates on each label and identify cases of misalignment scored between 0 and 1.

Fluency. Fluency refers to the naturalness of a text. One straightforward way to quantify fluency is by inverting the perplexity score output by a suitably strong yet compact language model like GPT-2 [250]. Intuitively, perplexity measures how well a language model can predict the next word in a given sequence of text. A lower perplexity score means that the text is not surprising and follows a similar distribution to the training data. Higher perplexity scores, on the other hand, suggest that the text is unexpected and unnatural. We adapt this criteria from prior work that used perplexity to judge the “plausibility” of adversarially transformed text [211]. Mathematically, the fluency of a text W can be calculated via

$$\begin{aligned} \text{Fluency}(W) &= 1/\text{Perplexity}(W) \\ &= 1 / \sqrt[N]{\frac{1}{P(w_1, w_2, \dots, w_N)}} \end{aligned} \tag{5.2}$$

where $P(w_1, w_2, \dots, w_N)$ is the probability assigned to all words in W .

Grammaticality. Grammaticality refers to the degree to which a text adheres to the rules and conventions of grammar. It focuses on the correct structure, syntax, and usage of words and phrases within a text. Grammatical errors, such as incorrect verb tense, subject-verb agreement, or sentence

fragments, can significantly affect the text comprehension. Unlike fluency, which emphasizes the naturalness and coherence of the text, grammaticality specifically assesses the accuracy of the language in terms of its adherence to grammatical rules. By distinguishing between fluency and grammaticality, we can address both aspects of language quality: ensuring the augmented text flows naturally while also maintaining proper structure. We quantify grammaticality with `language-tool` [167] which counts the number of grammatical errors present in the text, and invert this value to normalize the direction of improvement similar to fluency (i.e. higher is better). Formally, we define grammaticality of a text W via

$$\text{Grammaticality}(W) = \frac{1}{\text{Errors}(W) + 1} \quad (5.3)$$

where $\text{Errors}(\cdot)$ denotes the number of grammatical errors in W and the $+1$ prevents division by zero.

5.4 Experimental Setup

5.4.1 Benchmark Datasets

We conduct our experiments on four popular benchmark datasets covering sentiment analysis in SST-2 [282] and IMDB [198] as well as topic detection in AG News and Yahoo! Answers [347]. Dataset statistics are available in Appendix 5.8.1.

Following previous work [258, 316, 49], we explore the performance of all data augmentation strategies in a more challenging, *low-resource* setting. For each dataset, we constrain the number of labeled instances by subsampling only 10 examples per class. These subsets are expanded by $3\times$ for each baseline technique and our FADA approach.

5.4.2 Baselines

We compare our feature-aware data augmentation with the following baseline techniques:

Easy Data Augmentation (EDA) [316]. EDA is a fixed, data-set level augmentation framework

that uniformly samples one of four edit operations — synonym replacement, word insertion, word swapping, and word deletion — on p percent of words in a text. We use all the defaults of their open source implementation.

Checklist [261]. Initially proposed for testing NLP models, many of Checklist’s label-invariant transforms can actually constitute another fixed, dataset-level augmentation policy. We use five edit operations — replacing names, locations, and numbers, as well as contracting or expanding contractions — as implemented in TextAttack [213].

Text AutoAugment (TAA) [258]. TAA is an automated, dataset-level augmentation framework that learns to construct sequences of $\langle t, p, \lambda \rangle$ operations, where t is one of five transform $\in \{\text{Random Swap}, \text{Random Delete}, \text{TFIDF-Insert}, \text{TFIDF-Substitute}, \text{WordNet Substitute}\}$, p is the probability of t being applied, and λ is the proportion of words to be edited. New policy construction requires significant computational resources, so we only compare against the presearched policies the authors have open sourced for SST-2 and IMDB.

Uniform20. This augmentation strategy uses the same 20 transforms as FADA, but samples them uniformly, regardless of the features present in the text. Unlike other baselines that use different sets of transforms, Uniform20 specifically highlights the additional benefits of augmenting in a feature-aware fashion. Please refer to Appendix 5.8.2 for the detailed descriptions of each transform.

5.4.3 Text Classification Models

We run our experiments on two popular text classification models: BERT-base [71] and BERT-tiny [300, 23] as implemented in HuggingFace [321]. BERT-tiny was selected to highlight the impact of augmentation on more compact models commonly deployed to billions of edge devices. See Appendix 5.8.3 for details on the model and training routine.

5.5 Experimental Results

5.5.1 Augmentation’s Impact on Text Quality

To answer **RQ1.**, we study the effect different augmentation approaches have on improving or degrading the quality of text data. For comparison purposes, we add a Dist-2 [174] diversity metric, which measures the number of distinct bi-grams. For each baseline and dataset, we transform the texts and then compute alignment, fluency, grammaticality impact scores. These scores are normalized into ratios where values greater than 1 indicate improvements to that quality dimension relative to the original dataset and vice versa. For FADA, we performed a grid-search over quality weights — w_a, w_f, w_g — to generate 10 different augmentation policies (and datasets) and average the resulting scores throughout the rest of this work.

Table 5.1 shows the relative impact each augmentation approach had on our studied quality metrics across all datasets. With few exceptions, all augmentation frameworks tended to decrease text quality. For alignment and fluency, non-AutoDA approaches like EDA and CheckList score more highly, indicating that they better preserve the original meaning and naturalness of the text. This result may be explained by the fact that their underlying transforms are limited to making smaller edits less capable of injecting as much lexical diversity into the datasets. In contrast, FADA and Uniform20 exhibit significant diversity-quality tradeoffs.

Approach	Align.	Fluency	Gram.	Dist-2
Original	1	1	1	3372
EDA	1	1.02	0.83	3990
Checklist	1	0.95	0.82	3850
TAA	0.93	0.15	1.12	3996
Uniform20	0.94	0.44	0.55	5111
FADA (Ours)	0.91	0.46	0.58	5089

Table 5.1: Relative impact different augmentations have on our studied text quality metrics & Dist-2 as a comparative diversity metric, averaged across all datasets.

These results represent a starting point for understanding the relationship between data quality and model generalization. Next, we study the impact that these differences in quality have on post-train performance across four text classification tasks.

5.5.2 Generalization Performance

In order to address **RQ2.**, we use the datasets generated for **RQ1** to train 116 models between our two BERT variants. Table 5.2 shows the test F1-score of FADA and all baselines on four datasets. With few exceptions, all augmentation approaches yielded significant improvements over merely training on the original dataset. Only for BERT-tiny and SST-2 did most approaches fail to improve generalization while FADA was still helpful. Overall, our FADA approach achieved the highest F1-score on all four datasets for BERT-tiny. With an average boost of **11.7%**, this highlights the fact that lower-capacity models are especially good candidates for feature-aware augmentation. For the BERT-base model, FADA achieved the highest F1-score on three out of four datasets with an average boost of **7.7%**.

Model / Approach	SST-2	IMDB	AG News	Yahoo! Answers
BERT-tiny	39.9 $\pm$ 7.2	35.7 $\pm$ 4.7	31.5 $\pm$ 1.7	16.6 $\pm$ 0.2
+ EDA	37.6 $\pm$ 5.7	36.4 $\pm$ 4.1	59.8 $\pm$ 4.8	30.6 $\pm$ 2.1
+ CheckList	36.9 $\pm$ 4.1	40.6 $\pm$ 3.7	59.6 $\pm$ 0.1	29.2 $\pm$ 1.7
+ TAA	37.3 $\pm$ 9.2	40.4 $\pm$ 8.8	–	–
+ Uniform20	36.6 $\pm$ 9.2	45.2 $\pm$ 13.2	57.5 $\pm$ 0.6	30.1 $\pm$ 1.7
+ FADA (Ours)	42.3 $\pm$ 6.0	50.8 $\pm$ 6.8	59.8 $\pm$ 2.8	31.1 $\pm$ 1.5
BERT-base	42.9 $\pm$ 9.4	53.7 $\pm$ 3.3	71.1 $\pm$ 5.15	56.9 $\pm$ 1.1
+ EDA	49.6 $\pm$ 8.2	53.5 $\pm$ 7.5	77.6 $\pm$ 2.75	59.6 $\pm$ 2.1
+ CheckList	45.3 $\pm$ 8.2	52.5 $\pm$ 7.5	77.6 $\pm$ 3.81	60.9 $\pm$ 1.5
+ TAA	57.4 $\pm$ 2.2	57.7 $\pm$ 4.6	–	–
+ Uniform20	54.2 $\pm$ 3.4	56.9 $\pm$ 9.3	79.9 $\pm$ 0.25	61.5 $\pm$ 1.4
+ FADA (Ours)	58.1 $\pm$ 9.9	58.6 $\pm$ 5.4	77.4 $\pm$ 2.30	61.3 $\pm$ 1.2

Table 5.2: Test F1-score (%) with standard deviation of different augmentation approaches in a low-resource regime. Results are averaged across three runs. All approaches use a $3\times$ augmentation multiplier for fairer comparison.

5.5.3 Correlations Between Text Quality and Post-Train Generalization

Despite FADA-augmented datasets scoring amongst the lowest on our text quality study, models trained upon them end up being the most performant. To better understand this puzzling result, we conduct a correlational study (**RQ3.**) following the same routine described in Section 5.2. The only difference is that instead of measuring inference-time performance of models trained on the

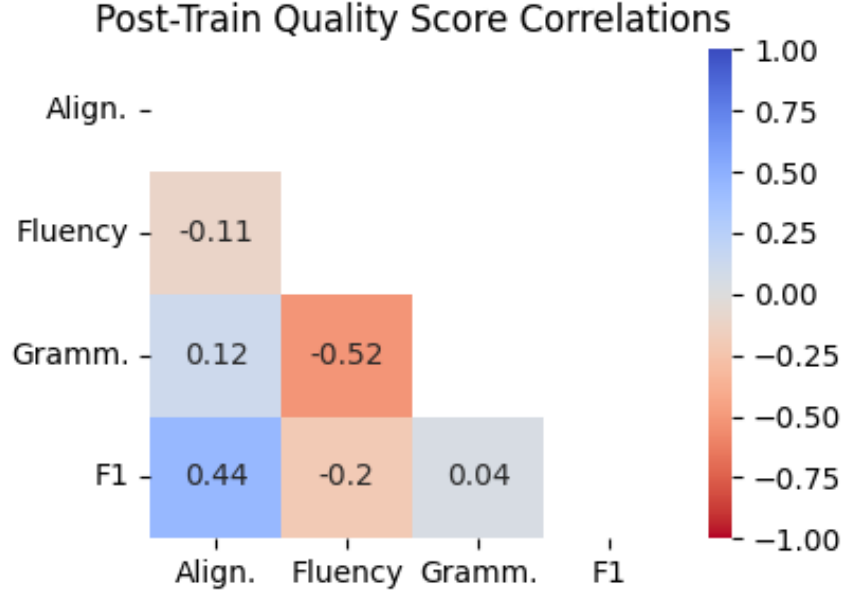


Figure 5.3: Quality metric correlations between each other and test F1-scores of models trained with augmented data, but tested on unagumented data.

original datasets, we used the models fine-tuned upon the augmented texts.

Figure 5.3 shows the correlations between our studied quality metrics and the F1-scores we use to quantify model generalization. Consistent with the combined findings for **RQ1.** and **RQ2.**, we observed weakly negative correlations for fluency and essentially no correlation between grammaticality and post-train performance. Only alignment enjoyed a moderately positive correlation of 0.44.

These results sit in sharp contrast to the preliminary correlational study used to initialize the relative importance weights that guided FADA to select useful transforms in the first place (Section 5.2). In that setting, the correlations characterized the impact text quality had on *inference* performance (i.e. the models were not trained on augmented data, only tested on it). Whereas this set of correlations characterize their impact on *training generalized models*, i.e. the models were trained on the augmented data and evaluated upon the original test sets. This provides strong evidence for the conclusion that text quality is significantly less important during training than it is for testing. However, the strong generalization results from **RQ2.** suggest that text quality combined with feature-awareness still constitutes an exceptionally effective strategy for judging the usefulness of augmentations.

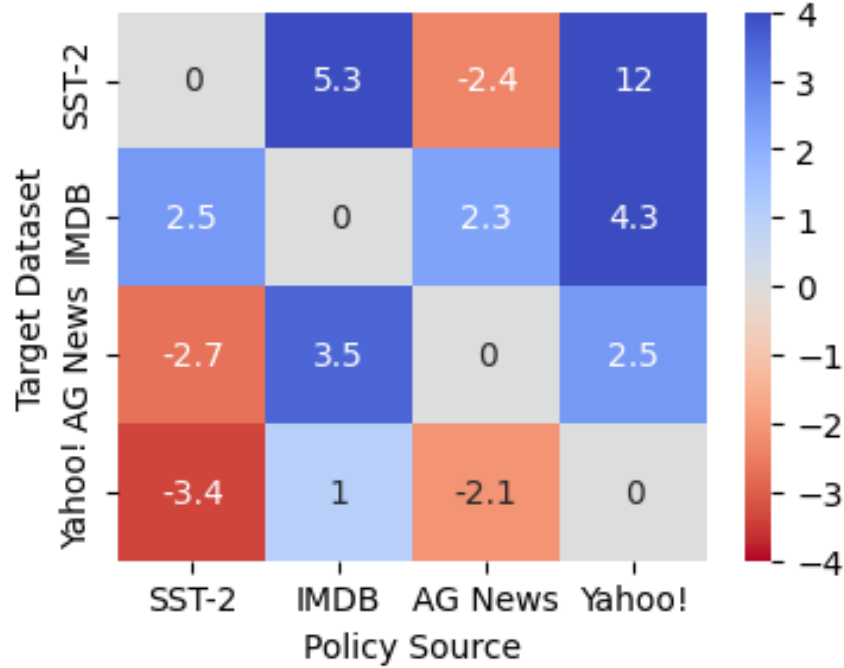


Figure 5.4: Transferability of FADA policies learned from a source dataset and applied to other target datasets. The numbers denote changes in F1-scores, averaged across both model architectures.

5.5.4 Transferability of FADA Policies

Finally, we conducted experiments to study whether the policies learned on one source dataset could be successfully applied to other datasets. As illustrated in Figure 5.4, policies learned for larger datasets with longer texts, such as Yahoo! Answers and IMDB, were able to make significant improvements when applied to datasets with shorter texts. These improvements even generalized across task domains (i.e. topic classification to sentiment analysis), with the most significant boost of 12.2% observed for SST-2 datasets augmented using the policies learned from Yahoo! Answers.

5.6 Related Work

Automated Data Augmentation. In recent years, there has been a growing interest in learning automated data augmentation policies for NLP tasks [331]. Such policies are a probability distribution over transforms according to which training samples are altered. Especially effective transforms are assigned higher probabilities and harmful transforms are assigned values near zero.

In an early work, [222] adapted AutoAugment [62] to discover effective augmentation policies for NLP tasks like dialogue generation. While their approach inherits the computational complexities of their predecessor, they are among the first to introduce instance-level augmentation policies conditioned on the text. This insight parallels our own about transform efficacy being a function of text features.

In addition to this work, there has also been research into dynamic policy scheduling for text augmentation. In this context, scheduling refers to the use of flexible policies that change over the course of model training. [176] found that dynamic policies could help improve text classification and machine translation performance. [55] also explored scheduling with a wider set of transforms and their results suggest that, while scheduling is helpful in some settings, the improvements were unsubstantial. Since scheduling currently exhibits empirically mixed results in the NLP domain, we simplified our approach to produce a stationary policy with an extra degree of freedom for features rather than time.

Text Quality and Generalization. Several related works applying augmentation to machine translation [241, 80] have also noted that better data quality did not necessarily lead to stronger models. [241] suggested that “lower-quality but *more diverse* data often yielded stronger results.” Acceptable trade-offs between quality and diversity suggest the existence of “sweet spots” within metrics that lie below the maximum possible scores. One concrete example of this can be seen in augmentation, helping to align training and testing distributions. For example, a perfectly grammatical training set will not be as helpful to model training if tested on text riddled with grammatical errors. Indeed, this idea is echoed in works like Fast AutoAugment [179] that use density matching to select augmentations that push the training data to more closely resemble the validation data.

5.7 Conclusion

In this work, we help characterize the complicated relationship text quality has on model generalization and show that alignment, fluency, and grammaticality are either moderately or strongly correlated with test-time inference performance. We combined this insight with the observation that transform efficacy is dependent on characteristics of the input text. Consequently, we designed

a novel, feature-aware data augmentation framework that closely monitors transform-feature interactions quantified by these text quality metrics. This approach to data augmentation opens up a new degree of freedom to learn more nuanced policies that slightly improve text quality relative to uniform sampling, but can significantly improve post-train F1-scores in a low-resource setting. Finally, an additional correlational study comparing augmented data quality to post-train generalization provides empirical support for the position that training on a larger amount of low-quality, but diverse data is helpful for acquiring performant NLP models.

5.8 Appendix

5.8.1 Benchmark Dataset Statistics

Table 5.3 shows various statistics for the datasets used in our experiments. Note that for SST-2, the test labels are officially hidden and scores can only be attained by submitting to the GLUE [306] benchmark. As a workaround, we evenly split the validation dataset and use the latter half for testing.

Dataset	Source	Task	Subject	Classes	# Train	# Test	Avg Len
AG News	[347]	Topic	News Articles	4	25,000	3,800	38
Yahoo! Answers	[347]	Topic	QA Posts	10	1,400,000	60,000	92
SST-2	[282]	Sentiment	Movies Reviews	2	70,000	436	18
IMDB	[198]	Sentiment	Movies Reviews	2	25,000	25,000	234

Table 5.3: Dataset statistics.

5.8.2 Studied Transform Descriptions

We conducted our policy search over 20 transforms implemented in the SIBYL augmentation tool [109]. Table 5.4 shows brief descriptions of each transform.

Transform Name	Description	Source
AddNeutralEmoji	Appends a random emoji with neutral sentiment	1
RemoveNeutralEmoji	Removes all emojis judged to exhibit neutral sentiment	1
ChangeHypernym	Randomly replace words with less specific words (e.g. turban → headwear)	2
ChangeHyponym	Randomly replace words with more specific words (e.g. fruit → apple)	2
ChangeLocation	Randomly change city and country names	2
ChangeName	Randomly change integers to other integers within 20% of the original	2
ChangeNumber	Randomly change names with some other names	2
ChangeSynonym	Randomly replaces words with approximate equivalents	2
ContractContractions	Contracts expanded contractions in a sentence (if any)	2
ExpandContractions	Expands contractions in a sentence (if any)	2
HomoglyphSwap	Replaces English characters with visually similar homoglyphs	3
RandomCharDel	Randomly deletes characters	3
RandomCharInsert	Randomly inserts characters	3
RandomCharSubst	Randomly substitutes characters	3
RandomCharSwap	Randomly swaps two adjacent characters	3
RandomInsertion	Randomly inserts a synonym of some word to a new position	3
RandomSwapQwerty	Randomly swaps characters with others adjacent on a QWERTY keyboard	3
InsertPunctuationMarks	Randomly inserts various punctuation marks	4
RandomSwap	Randomly swaps two adjacent words	5
WordDeletion	Randomly deletes words	5

Table 5.4: Descriptions of all the text transforms in the FADA search space. Sources: 1. SIBYL [109] 2. CHECKLIST [261] 3. TEXTATTACK [213] 4. AEDA [136] 5. EDA [316]. Note that source attributions are based on implementation details, not necessarily where the transformation was initially proposed.

5.8.3 Model and Training Settings

BERT-base (`bert-base-uncased`) and BERT-tiny (`prajjwall/bert-tiny`) are two different sizes of the same encoder-only BERT architecture. BERT-base has 12 transformer layers, a hidden representation size of 768, feed-forward size of 3072, and 12 attention heads. On the other hand, BERT-Tiny has 2 transformer layers, a hidden representation size of 128, and an unspecified feed-forward size and number of attention heads. Smaller BERT models like BERT-Tiny are intended for environments with restricted computational resources, like cellular devices. They can be fine-tuned in the same manner as the original BERT models.

We used HuggingFace [321] to fine-tune our models on the training dataset. We made heavy use of the best practices encoded in the defaults of their `Trainer` class, though with several customized settings. We set the initial learning rate to 0.001 with a weight decay of 0.01. The batch size for training and evaluation were set to 4 and 16, respectively. The training process was run for a maximum of 10 epochs, with evaluation performed at the end of each epoch and early stopping if validation loss could not be improved after 5 epochs. The best performing model checkpoint was saved and used for evaluation on the test dataset.

CHAPTER 6

Human-in-the-Loop Synthetic Text Data Inspection with Provenance Tracking

The Feature-Aware Data Augmentation (FADA) framework offers a sophisticated approach to navigating the complex landscape of text augmentation. By integrating text quality metrics and semantic features, FADA demonstrates how to algorithmically learn effective policies that balance the generation of diverse data with the preservation of linguistic quality (RQ3). This represents a significant step towards automating the creation of high-quality augmented datasets, particularly in low-resource scenarios. However, despite these algorithmic advances, the inherent complexity of language means that automated quality assessment is not foolproof. Metrics may fail to capture subtle semantic shifts, and even the most carefully optimized policies can occasionally produce flawed, nonsensical, or mislabeled outputs that could negatively impact downstream model training or evaluation.

Recognizing that purely automated solutions may still fall short, especially in sensitive applications requiring high data fidelity, the need for effective human validation persists. Manually sifting through vast amounts of synthetic data is impractical, demanding tools that can make human oversight both efficient and targeted. The INSPECTOR system, presented next, directly addresses this challenge. It builds upon the understanding that different transformations introduce different types of potential errors by leveraging transformation provenance. By enabling human reviewers to group and analyze synthetic texts based on their generative origins (RQ3), INSPECTOR facilitates a structured, scalable approach to identifying and correcting flaws in augmented datasets, bridging the gap between automated generation and human-verified quality assurance.

<u>Original</u> <u>text</u>		<u>Transformed</u> <u>text</u>	<u>Label</u>	<u>Is</u> <u>label</u> <u>correct?</u>
ends up being surprisingly dull	Word Deletion	up being surprisingly	-	👎
the rich promise of the script will be realized on the screen		the rich will be on screen	+	👎
forget about it by Monday	Change Hypernym	omit about it by Monday	-	👎
the movie is beautiful to behold		the event is beautiful to see	+	👍

Figure 6.1: Examples of transformed texts from the SST2 movie review dataset generated during data augmentation. A transformed text can contain garbled text, or have an inappropriate label. As an example, the “Word Deletion” transformation can mangle the text “ends up being surprisingly dull” into “up being surprising”, causing its corresponding label “-” (indicating a negative sentiment) to no longer be appropriate. Of the four examples of synthetically generated texts, only one (“the event is beautiful to see”) has an appropriate label.

6.1 Introduction

Data augmentation techniques to generate additional training data by transforming existing data can help improve model performance and robustness. However, low-quality texts with garbled text and inappropriate labels may be generated. Figure 6.1 shows examples of high- and low-quality instances after a transformation is applied. Despite users’ inclination to filter out texts of low quality with inappropriate labels, effective debugging of the generated content remains challenging due to the opaqueness of these techniques and the sheer volume of data produced. Investigating the data instances one by one would be extremely demanding and slow.

We propose a human-in-the-loop approach, INSPECTOR, for inspecting generated texts to weed out texts with incorrect labels. For reducing human effort, INSPECTOR applies provenance tracking, inspired by work in the database community [310], and assistive labeling. INSPECTOR supports analysis of the provenance of each text in two ways. First, INSPECTOR allows users to group the texts by their *transformation provenance*, i.e., the common transformations that have been applied to produce the text. Second, INSPECTOR allows texts to be grouped by their *feature provenance*, i.e., common linguistic features, e.g., if the text contains a negation, obtained from the

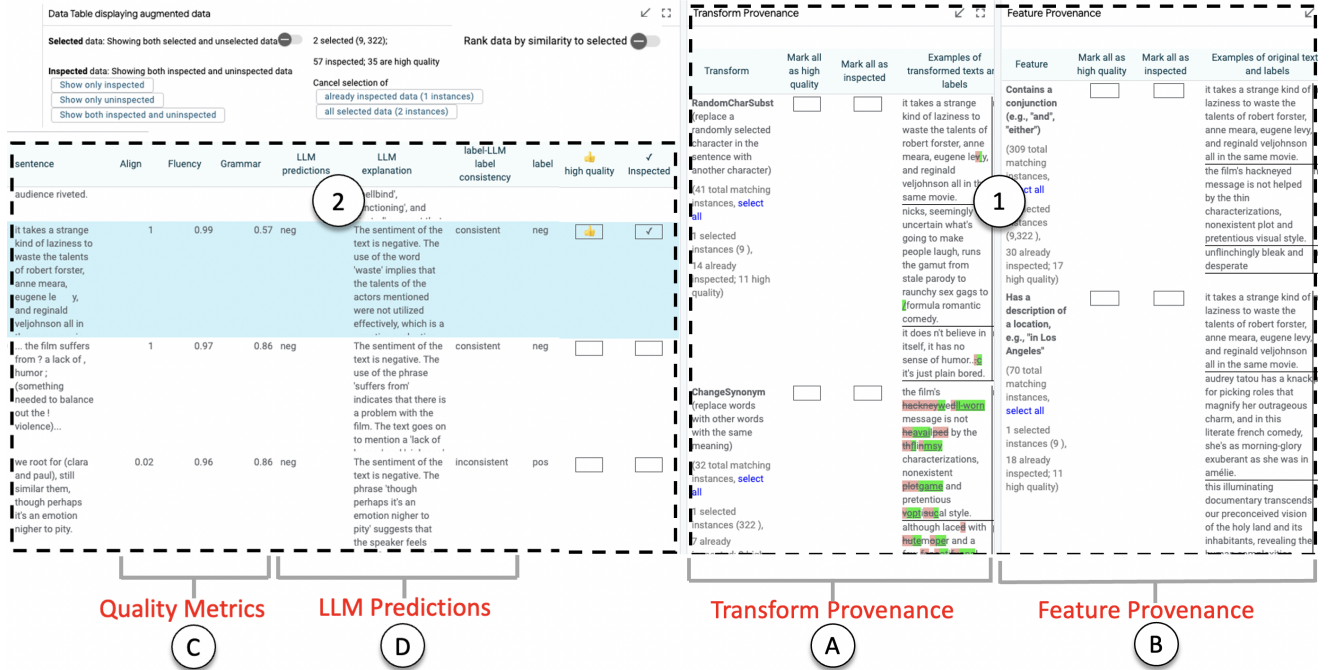


Figure 6.2: INSPECTOR: The user alternates between (1) inspecting the provenance of groups of texts and labels following their (A) common transformation, and (B) common linguistic features, and (2) inspecting individual transformed texts with their corresponding labels, with assistive labeling using (C) the quality metrics, alignment, grammaticality, fluency scores, and (D) LLM predictions.

relations represented in Abstract Meaning Representation graphs [14]. Provenance tracking allows the inspection of groups of texts with the same applied transformations or underlying feature. For example, the texts in Figure 6.1 transformed by *WordDeletion* share a common transformation and can be grouped together by INSPECTOR.

For assistive labeling, INSPECTOR provides two techniques. INSPECTOR computes quality metrics, such as label alignment, grammaticality, and fluency, for each generated text and corresponding label. Finally, INSPECTOR provides the predictions from a large language model that users can compare against the corresponding labels of the texts to find discrepancies.

To evaluate INSPECTOR, we ran a within-subject user study with 15 participants. The participants weeded out generated texts with inappropriate labels on two datasets, a sentiment analysis dataset [283] and a hate speech detection dataset [17]. We build a baseline by disabling the provenance tracking and assistive labeling features. Participants using INSPECTOR identified 3x and 4x more texts with correct labels (Welchs t-test: $p < 0.005$). Using INSPECTOR, participants

were more confident in identifying texts with correct labels, and adopted systematic inspection strategies. The human-selected texts and labels improve model robustness more than randomly sampled data, demonstrating the value of human inspection. No single technique of INSPECTOR was useful to every participant, suggesting that effective inspection of generated texts requires combining complementary techniques.

In summary, INSPECTOR is an approach for inspecting generated texts and corresponding labels using a novel technique for grouping texts by their provenance. INSPECTOR also offers assistive labeling techniques. Our tool is open source [301]. A within-subject user study shows that using INSPECTOR enables more effective inspection and that users found grouping texts by their transformation provenance to be the most useful feature. The human-inspected data improves model robustness by up to 32%. We find that no single technique, including LLM-based assistive labeling [95, 311], takes away the need for human inspection of generated texts.

6.2 Design Goals and System Overview

6.2.1 Design Goals

Acquiring data is a bottleneck for machine learning [232], but data labeling is not a simple task [108, 159, 43]. In particular, ensuring the quality of data is critical [178, 269, 232]. We identify challenges for inspecting generated texts:

Scale. Data annotation aims to obtain as much data at the lowest possible cost [308]. Inspection of generated texts shares a similar goal: INSPECTOR should empower users to identify large subsets of texts with correct labels.

Evaluating each instance. Human inspection is difficult because the data-generating processes are opaque. Moreover, users may not enjoy trivial labeling tasks [37] and may clean data without rigor [154]. Hence, INSPECTOR should discourage ad-hoc inspection and respect human cognition by providing more support for analyzing each text and discovering systematic insights.

Transform	Mark all as high quality	Mark all as inspected	Examples of transformed texts and labels
RandomCharSubst (replace a randomly selected character in the sentence with another character) (41 total matching instances, select all) 1 selected instances (9), 14 already inspected; 11 high quality)	☐	☐	it takes a strange kind of laziness to neg waste the talents of robert forster, anne meara, eugene le^{ly}, and reginald veljohnson all in the same movie. nicks, seemingly uncertain what's neg going to make ^{people} laugh, runs the gamut fr^{om} ale parody to raunchy sex gags to ^{formula} romantic comedy.

Figure 6.3: Transform provenance. A user selects texts and inspects the common transforms (e.g., RandomCharSubset) in the transformation provenance pane with their (A) inspection statistics (e.g., the user has inspected 14 texts, with 11 marked as high quality), and (B) view other texts sharing the same transform. A user can then (C) mark all instances sharing the same transform to be correct, obviating the need for inspecting individual texts one by one.

6.2.2 Overview

Workflow. INSPECTOR supports the workflow shown in Figure 6.2. Through INSPECTOR, users inspect the generated texts and identify texts with correct labels, which are retained in the dataset. INSPECTOR enables inspection of the provenance (the two panes (2) in Figure 6.2, Section 6.3.2), and assistive labeling (on each generated text and corresponding label (the table (1) in Figure 6.2, Section 6.3.3). For grouping generated texts using provenance tracking, INSPECTOR offers information about (1) the transformations applied to the selected data in the Transformation Provenance pane, and (2) the linguistic features present in the text before transformations were applied in the Feature Provenance pane.

For assistive labeling, INSPECTOR provides (1) computed quality metrics (i.e. grammaticality, fluency, and label alignment) for each instance, and (2) the predictions of a large language model.

We envision that a user of INSPECTOR alternates between the inspection of common transformations and features, forming hypotheses about root causes of quality, and the inspection of individual instances with the help of assistive labeling.

Feature	Mark all as high quality	Mark all as inspected	Examples of original texts and labels	
Has a description of a location, e.g., "in Los Angeles" (70 total matching instances, select all) 1 selected instances (9), 24 already inspected; 11 high quality)	☐	☐	it takes a strange kind of laziness to waste the talents of robert forster, anne meara, eugene levy, and reginald veljohnson all in the same movie.	neg
			audrey tatou has a knack for picking roles that magnify her outrageous charm, and in this literate french comedy, she's as morning-glory exuberant as she was in amélie.	pos

Figure 6.4: Feature provenance. A user can select texts, and can inspect linguistic features common to the selected texts (e.g., “Has a description of a location”) in the transformation provenance pane with their inspection statistics (e.g., the user has inspected 24 texts, with 11 marked as high quality). Then, a user can mark all instances sharing the same feature to be correct.

6.2.3 Usage Scenario.

Suppose that Alice is a model developer who wishes to expand a training dataset. Alice turns to data augmentation but is cynical. She previously observed that these techniques produce large quantities of data with the majority of texts garbled or have unsuitable labels. Alice would not trust a model trained with poor quality data (after all, *garbage in, garbage out*). As she wishes to retain only texts with correct labels but finds going through all instances onerous, Alice gives INSPECTOR a try.

Alice provides the original training dataset to INSPECTOR. INSPECTOR applies the data augmentation techniques to generate data. Next, Alice inspects the texts (Figure 6.2). First, she inspects several individual instances. As she marks their quality, she hypothesizes that the quality of the texts are influenced by a linguistic feature (e.g., in sentiment analysis, deleting a single negation feature in “I do **not** like stand-up.” inverts the sentiment of the text), or a transformation that performs only small-scale modifications. To assess these hypotheses, she selects several instances of interest. Now, the provenance panes display information about the transformations and

underlying features of the selected instances (Figure 6.3).

Alice analyzes the common transformations and underlying linguistic features. Each common transformation and feature has examples of data sharing the same provenance pattern, enabling Alice to understand the transformation or feature. INSPECTOR tracks and displays inspection statistics such as the total number of inspected records and the proportion of records annotated as having suitable labels. As Alice continues to mark data, these statistics are updated. The frequent updates allow Alice to establish trust that the information displayed is meaningful [170, 79]. From the small proportion of texts with correct labels among the inspected data (in Figure 6.4), Alice dismisses her hypothesis regarding the linguistic feature. However, the same statistic for a common transform in the transformation provenance pane (Figure 6.3) appears to support the hypothesis regarding a transformation that rarely distorts the text.

Alice deepens her investigation. She filters the generated texts (Figure 6.6) to show only texts sharing the common transformation, but there is still too much data. Alice reorders the texts by their grammaticality. She finds that even the lowest-scoring instance has a sufficiently high score (e.g., score > 0.8), suggesting that the texts are never too distorted. Skimming through the texts, she notices that the majority of them have labels that are consistent with the large language model's predictions, suggesting that the transformation usually preserves semantics. With her hypothesis validated, she marks all instances in the group. She notices that some instances with incorrect labels have poor fluency scores. She reorders the texts by their fluency and unmarks the instances with low fluency scores.

Having identified a strategy of finding common transformations among high-quality instances and assessing the grouped instances with the large language model's predictions and quality scores, Alice continues until she believes she has enough data for training the model.

6.3 Implementation

6.3.1 Overview

Figure 6.5 shows an overview of how a user interacts with INSPECTOR. When a user has selected several texts, INSPECTOR displays both the synthetically generated texts, as well as their common transformations and features (Section 6.3.2). For each individual text instance, INSPECTOR computes quality metrics and displays the predictions of a large language model (Section 6.3.3).

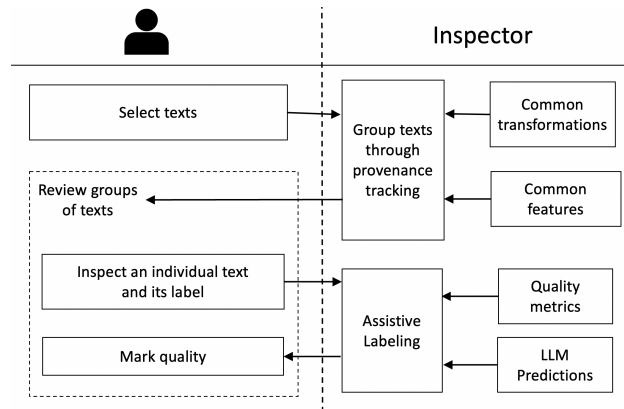


Figure 6.5: The workflow of a user inspecting data using INSPECTOR.

6.3.2 Provenance Tracking

The Transform Provenance and Feature Provenance Panes (Figure 6.3 and Figure 6.4) surfaces common transformations and linguistic features in the provenance of the user-selected generated texts. Providing details about the transforms and features, INSPECTOR allows users to inspect groups of texts sharing either the same transformations or linguistic features.

The Transformation Provenance pane summarizes recurring patterns in the transformations applied to generate the data. This allows the investigation of root causes of poor quality related to the text transformations, e.g., a random word deletion. The Feature Provenance pane displays common linguistic features (extracted from Abstract Meaning Representation graphs [14]) in the texts before they were transformed. This allows the investigation of possible root causes of low quality stemming from features, e.g., negations (“not”), in original texts.

A				B			
sentence	Align 	Fluency	Grammar	LLM prediction	LLM explanation	label-LLM label consistency	label
true fairytale of valourousness -- and complicity -- at auschwitz is a harrowing tragedy that push to unbosom of the	0.79	0.98	0.81	neg	The sentiment of the text is negative. The use of words like 'harrowing tragedy' and 'unspeakable' suggest that the	inconsistent	pos
fun}ny but perilously slight.	0.8	0.85	0.86	neg	The sentiment of the text is negative. The use of the word 'perilously'	inconsistent	pos

Figure 6.6: INSPECTOR enables assistive labeling by providing (A) quality metrics such as label alignment, fluency, and grammaticality. Label alignment is a measure of label quality, while fluency and grammaticality are measures of linguistic quality. It also shows the (B) LLM predictions for a text compared to its corresponding label.

The inspection statistics are presented for each common transform or feature (in the example in Figure 6.3, the user has inspected 14 texts in total, with 11 of them marked as high quality), enabling the user to make generalizations about the group of data. Examples of other instances in the group are presented, with the transformed parts of the text highlighted to focus human attention, which may increase labeling efficiency [54]. Users can filter data to view other instances with a shared transformation or feature provenance.

Batch inspection. Users can mark a batch of instances [7] on a group of data (e.g., all data produced by the same transformation type). This enables greater inspection efficiency by applying the same decision across the group.

6.3.3 Assistive labeling

Figure 6.6 shows the table containing the transformed texts and their corresponding labels. INSPECTOR presents the following quality metrics:

1. Fluency, measured through language perplexity using GPT-2 [250],

Table 6.1: The text transformations for generating data in the user study. For example, given the text “ends up being surprisingly dull” with a “negative” label, “Word Deletion” produces a new text “up being surprising” with the same corresponding label.

Category	Transformation
Swap	ChangeHypernym, ChangeHyponym, ChangeLocation, ChangeName, ChangeNumber, ChangeSynonym, RandomSwap, RandomSwapQwerty
Punctuation	ContractContractions, ExpandContractions, InsertPunctuationMarks
Typos	HomoglyphSwap, WordDeletion, RandomCharDel, RandomCharInsert, RandomCharSubst, RandomCharSwap
Text Insert	RandomInsertion
Emojis	AddNeutralEmoji, RemoveNeutralEmoji

2. Grammaticality, measured as the degree to which the text is free from grammar errors using `language-tool` [167],
3. Label Alignment, measured through predictions of label quality [223].

While Fluency and Grammaticality are measures of linguistic quality, Label Alignment measures label quality using CleanLab [60], a method for identifying mislabelled data [223]. We normalize each score such that they range between 0 and 1, with 0 indicating the lowest quality and 1 the highest. INSPECTOR allows the texts to be sorted by these metrics.

INSPECTOR provides information about the outputs of a large language model (*gpt-3.5-turbo*) prompted to predict the transformed texts’ labels: 1) its predictions, 2) explanations of each prediction, 3) consistency of its predictions with the instances’ labels. Viewing predictions from the LLM, which may have human-level performance [95, 311], can increase labeling efficiency [164, 68] and explanations contextualizing each prediction increases the user’s trust in them [16]. Inconsistencies between labels and LLM predictions may imply altered semantics after a text transformation. Together with the quality metrics, the outputs of the LLM provide evidence for users to make decisions, guiding them away from ad-hoc assessments.

Table 6.2: The two datasets in the user study. We analyze only the synthetic texts generated using the data augmentation techniques. During the course of the user study, 100% and 99% of the SST2 and TweetEval datasets were reviewed by at least one participant.

Dataset	Description	Size
SST-2	Predict the sentiment of a movie review.	613
TweetEval (Hate)	Predict if a tweet contains offensive discourse.	763

6.3.4 Study Design

Tasks. The study involves two datasets, the sentiment analysis dataset, SST2 [283], and a hate speech detection dataset, TweetEval [17], described in Table 6.2. Table 6.1 shows the list of considered text transformations [215, 263, 137, 317, 110]. For each task completed by the user, we finetune BERT [70] on the identified high-quality data for up to 10 epochs.

Participants. We recruited 15 participants by reaching out to students in the Computer Science department. 11 of them are Ph.D. students, 2 are master students, and 2 are undergraduates. 7 participants had less than 1 year of machine learning experience, 2 had about one year, 2 had 2-5 years of experience, and 2 had more than 5 years of experience. The participants also self-reported their familiarity with inspecting machine learning datasets and understanding mislabelled data on a 7-point Likert scale. The mean familiarity was 3.4, where 1 is “Most unfamiliar” and 7 is “Most familiar”. This level of experience is identical to data annotators in industry, who do not have a machine learning background [308].

Study Protocol. Our study involves two datasets and two tools. We design a within-subjects study where each participant investigates both datasets and experiences using both tools. For each task, the participant used only the assigned tool, either INSPECTOR or Annotator, the variant of INSPECTOR without the effort reduction techniques. The study requires the completion of 2 tasks, with each task requiring 20 minutes at most. We design our study to be completed in 60 minutes. Before the users started on a task, we asked for the participants’ consent to record their usage of the tools. Then, they spent 10 minutes working through a tutorial and warm-up questions. After completing the tasks, the participants were directed to a post-study questionnaire to share their experiences and feedback about the tools. We also solicited responses about the participants’

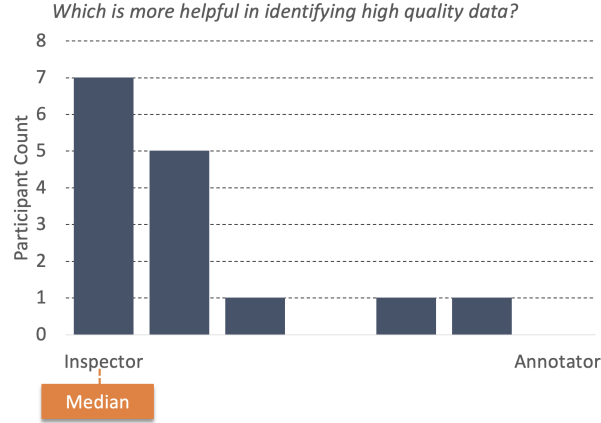


Figure 6.7: Most participants perceived INSPECTOR to be more helpful in winnowing out generated texts with incorrect labels (such as the examples in Figure 6.1).

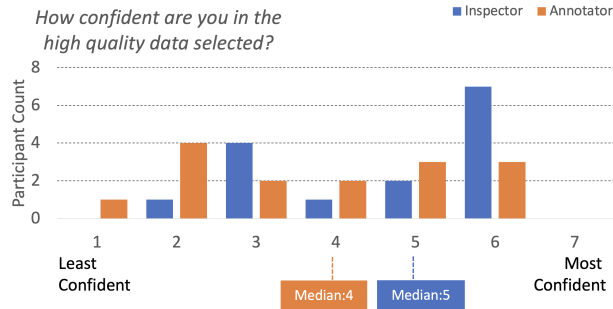


Figure 6.8: Participants using INSPECTOR were more confident of their inspections, with a median rating of 5 compared to the baseline of 4.

inspection strategy.

6.4 Results

In this section, we report and analyze the results of our user study. We denote each participant as P#.

6.4.1 Reduction in Human Effort

To assess inspection efficiency, we counted the number of texts with correct labels identified by the participants. Table 6.3 shows that using INSPECTOR leads to 3x and 4x more texts on the sentiment analysis dataset and the hate speech detection dataset. Using INSPECTOR, participants identified an average of 277 and 259 high-quality instances compared to 82 and 63 instances using

Table 6.3: The average number of texts with correct labels identified. INSPECTOR improves over a baseline without provenance tracking and assistive labeling by $3\times$ and $4\times$. SA: Sentiment Analysis, HS: Hate Speech Detection

Approach	SA	HS
INSPECTOR w/o provenance tracking and assistive labeling	82	63
INSPECTOR	277	259

the baseline on the SST2 and TweetEval’s Hate Speech dataset, respectively. The differences in efficiency is significant (Welch’s t-test, $p < 0.005$).

Figure 6.7 shows that the participants found INSPECTOR more useful and were more confident of their results. 13 of the 15 participants indicated that they preferred using INSPECTOR over the baseline tool. Participants had a higher median level of confidence in their inspections using INSPECTOR (5 vs 4 on a 7-point scale, in Figure 6.8).

The techniques in INSPECTOR changed the participants’ perception of the task’s nature. P3 indicated that INSPECTOR “could give initial results. I’m more acting like a verifier”. They found INSPECTOR more usable in identifying patterns. Comparing INSPECTOR and the baseline, 13 of the 15 participants found INSPECTOR more helpful in identifying patterns for inspecting the texts. They reported using the techniques of INSPECTOR in their inspection strategies. P8 wrote “My main strategy was to find inconsistent patterns in already labeled data and LLM” while P1 mentioned that “Sentences with a grammar score < 0.92 are almost always low-quality.”. Conversely, using the baseline, participants found strategically inspecting the data difficult. P8 wrote that dissecting the data “is utterly not possible” and “Using `ctrl+f` was really painful.”

6.4.2 User Ratings of Individual Features

In the post-study questionnaire, participants rated the techniques of INSPECTOR and described how they inspected the data. Table 6.4 summarizes the participants’ feedback. Grouping data by transformation provenance was the most appreciated technique, followed by the assistive labeling techniques. On a 7-point Likert scale, participants assigned Transform Provenance the highest average rating of 4.5, followed by the two assistive labeling techniques with average ratings of 4.3. P8 indicated that using transform provenance was the “main strategy I used to identify trends”.

Table 6.4: Usefulness of the techniques of INSPECTOR rated by the participants (out of 7). A higher rating indicates the technique was perceived to be more useful for inspection. Grouping texts by common transformations was perceived to be the most useful technique.

Technique	Average Rating
Transform Provenance	4.5
Quality Metrics (grammaticality, fluency, & alignment)	4.3
LLM Guidance	4.3
Feature Provenance	1.9

Participant C7 wrote that provenance tracking allows her to “reason about whether a specific transformation can lead to a reduction of data quality”.

Diverse inspection strategies. The responses to the post-study questionnaire reflected a wide range of strategies. No single technique of INSPECTOR was found to be useful by every participant. We qualitatively analyze the free responses. A majority (11) of the participants used the transformation provenance to make decisions. 9 participants described using the quality metrics (grammaticality, fluency, label alignment), and 8 participants mentioned using outputs of the large language model. All participants found at least one technique of INSPECTOR to be useful (rating at least 4 out of 7 and using it in their workflow). This suggests no one technique obviates the need for human inspection.

Batch Inspection. A majority (11 of the 15) of the participants used the batch inspection feature, marking an average of 5 groups. The participants appreciated batch inspection and used it after manual validation of a few representative instances. P13 wrote “if most of the labels are not affected by the transformation and are indeed high-quality by manual inspection, I mark all such entries as high-quality.”. However, not all participants found grouping the texts by their provenance useful or meaningful. Two participants applied a strategy of trying to go through the data one by one. P11 wrote that she did not consider data provenance for grouping the texts “in order to give an unbiased opinion” for each text and its label.

Table 6.5: Results comparing the robustness of BERT after finetuning on additional data. We measure the attack success rate of DeepWord (the lower the better). SA: Sentiment Analysis, HS: Hate Speech Detection

Approach	SA	HS
Randomly sampled	0.61	0.50
Human-guided	0.59	0.34

6.4.3 Robustness

Next, we assess the improvements in model robustness when training a model with the texts with correct labels collected by the participants. As a baseline, we construct a randomly sampled dataset with the average number of data instances selected using INSPECTOR. We assess the robustness of the model by measuring the attack success rate of DeepWord [88], a method of generating adversarial attacks. Using the implementation of DeepWord in TextAttack [215], we generated 100 attacks on each model. More robust models would face fewer successful attacks.

On the SST2 dataset, 4 out of 8 participants marked data that led to more robust models than randomly sampled data. On the TweetEval dataset, all 7 participants identified data that led to more robust models than randomly sampled data. The attack success rate of DeepWord on models trained with randomly selected data was 0.61 on the SST2 dataset and 0.5 on the TweetEval dataset. Using the inspected data, the attack success rate decreases to an average of 0.59 and 0.34 on SST2 and TweetEval, respectively. On TweetEval, this corresponds to a 32% improvement. Overall, the texts identified using INSPECTOR improved model robustness.

6.5 Implications

Our study showed that INSPECTOR empowered users to be effective and confident in inspecting transformed texts and their corresponding labels.

Transformation provenance was the most useful technique. The participants perceived transformation provenance to be the most useful technique provided by INSPECTOR. Even when participants did not use it to perform batch inspection, they found the additional information useful. P2 considered this information to inspect data, writing that it “showed whether a malformed

sentence was in the original text or mangled by a transformation.”.

Assistive labeling helped users build trust. The participants were aware of the risk of including incorrect labels when labeling a batch of texts. Thus, the users had to trust the guidance and automation provided by INSPECTOR. Users were able to build trust using the inspection statistics of the groups of generated texts. The assistive labeling techniques were also helpful for building confidence. P8 wrote “When the fluency and grammaticality scores are both high, I am more confident to label the data as high quality.”

The linguistic features were perceived to be ineffective. Surprisingly, users found grouping the texts by their common linguistic features to be ineffective. The participants did not always understand their relevance to the task. P14 wrote “the features are very low level and it is not clear how they are related to the labeling quality”.

Limited impact on inspection accuracy. We observed that users tended to only mark texts they were confident about. Comparing the participants’ inspections to the ground-truth labels annotated by one of the authors of this paper, we find that the use of INSPECTOR only had a limited impact on the accuracy of the user’s inspections. On SST2, INSPECTOR slightly decreased labeling accuracy from 88% to 85.3%. On TweetEval, INSPECTOR increased accuracy from 89.1% to 90.0%. These differences are not statistically significant ($p\text{-value} > 0.05$). This suggests that users of INSPECTOR inspected more texts with less effort and increased confidence while maintaining the same level of accuracy.

6.6 Related Work

6.6.1 Debugging Machine Learning

Testing models. DynaBench [140] evaluates models with challenging human-provided data. AdaTest [260] generates more test cases similar to user-provided tests. These studies focus on models but do not support debugging data, which is the focus of INSPECTOR.

Grouping data for data inspection.

SliceFinder [57] and What-If [320] are tools for understanding the subsets of data with poor

model performance. Zeno [35] groups data on properties extracted by user-written Python programs. Tempura [323] supports data inspection by grouping data using structural templates abstracted from concrete data instances. Unlike these tools, INSPECTOR groups data by their provenance.

6.6.2 Cleaning Data

For cleaning tabular data, Data Civilizer [259] is a data pipeline-debugger for identifying low-quality data (e.g., malformed values) that cause incorrect data analysis outputs through inserting breakpoints and the visualization of the intermediate records. Unlike Data Civilizer, INSPECTOR is not a breakpoint debugging tool. Instead, it helps users filter out data with unsuitable labels. Wrangler [133] cleans dirty data by inferring data types (e.g., integers) and semantic roles (e.g., zip code) for identifying anomalous data. Potters Wheel [253] cleans dirty data, but detecting them may require users to implement an API to define constraints (e.g., date formats). On the other hand, INSPECTOR assists human users in interactively identifying texts with incorrect labels without requiring programming literacy.

Ruler [53] and TagRuler [82] learn rules for annotating unlabelled data based on the text. INSPECTOR guides users to weed out texts with incorrect labels. While feature provenance in INSPECTOR is similar to the token-based rules in Ruler or TagRuler, it uses linguistic features rather than tokens.

6.7 Conclusion

We present INSPECTOR for winnowing synthetic texts with incorrect labels generated during data augmentation. In a within-subject user study, participants using INSPECTOR were $3\times$ and $4\times$ more effective. Users found that grouping data by their shared common transformations to be the most useful technique. Assistive labeling allowed them to build trust in the tool. Surprisingly, users perceived the linguistic features to be ineffective. INSPECTOR is the first interactive human-in-the-loop approach for examining text augmentation data for classification tasks by com-

binning provenance inspection and assistive labeling techniques.

CHAPTER 7

Measuring Psychological Depth in Language Models

The preceding sections have built a case for enhancing machine learning evaluation through sophisticated data augmentation and validation techniques. We began by demonstrating with SIBYL (Chapter 4) that expanding data-space coverage, even by transgressing traditional label-preserving constraints, significantly improves model robustness and generalization (RQ2). Subsequently, FADA (augmentation 5) introduced a method for optimizing this diversification process algorithmically, leveraging semantic features and quality metrics to navigate the complex trade-offs involved (RQ3). Recognizing the limits of automation, INSPECTOR (augmentation 6) then highlighted the necessity of human oversight, providing a provenance-aware framework for efficiently validating synthetic data quality (RQ3). Collectively, these contributions address the critical need for diverse, high-quality data to build and rigorously test more capable ML models, focusing primarily on the structural and semantic integrity of the data itself.

However, a truly holistic evaluation, particularly for AI systems generating creative or interactive content like narratives, must extend beyond the properties of the input data or the technical correctness of the output. While diverse and clean data (as pursued by SIBYL, FADA, and INSPECTOR) provides a stronger foundation for models, it does not guarantee that the generated output will resonate with human users or achieve its intended communicative purpose. We must also consider the subjective experience elicited by the AI’s creations. Qualities such as engagement, emotional impact, perceived authenticity, and narrative richness are crucial aspects of human-centric evaluation that remain largely unaddressed by traditional metrics or data-focused approaches. Therefore, the dissertation now shifts focus from the input side (data diversity and quality) to the output side (reader perception), introducing the Psychological Depth Scale (PDS) in the next section (Section 1.4). This scale provides a novel framework for systematically measuring

Premise: A psychic alien who feeds on dreams comes to Earth for the first time. Turns out humans are the only sentients in the galaxy that have nightmares.

Story Excerpt	
... In an apartment nestled within a labyrinth of city buildings, Elijah tossed and turned, shrouded in the dark veils of sleep. The horrors of a murky past danced upon his eyelids as nightmares consumed him repeatedly. His brains spun enigmatic codes during the day, and his unconscious mind spun the threads of terrifying stories at night. The tranquility of sleep was an alien sensation to him. However, something shifted that night as he slipped into uncharted serenity, the nightmares oddly dissipated. Transparent and silent, Nebula stood by his bed, gently feeding on his dreams. The sheer rawness of human fear sent shivers pulsating through her. The nightmares were unexpectedly addictive, an uncanny ecstasy that filled her with dreadful energy and intense guilt. She wrestled with the unsettling contrast between her survival and the pain she absorbed from Elijah's complex human mind. ...	
Psychological Depth Scale	
Emotion Provocation	2.6
Empathy	2.6
Engagement	4.2
Authenticity	3.4
Narrative Complexity	4.4
Coherence	5.0
Fluency	5.0
Toxicity	0.0
Bias	0.0

Figure 7.1: In this GPT-4 story, the psychological depth scale highlights strengths and weaknesses contributing to the overall reader experience, providing additional quality signals over traditional metrics more likely to saturate. Scores are normalized 1-5 for comparison.

these vital subjective and experiential dimensions (RQ4), complementing the previous work by adding a crucial layer of qualitative, human-centered assessment to our evaluation toolkit.

7.1 Introduction

Stories play a crucial role in our understanding of ourselves and the world around us [166, 134]. As large language models (LLMs) are increasingly deployed in narrative design and creation, their growing impact on how stories are told calls for a deeper understanding of their narrative power. Current evaluations of LLM-generated stories often focus on objective properties of the text such as discourse structure [189], fluency [98], style [312], creativity [41], diversity [91], toxicity and bias [307]. However, it is crucial to extend these evaluations to accommodate the subjective, psychological impact stories have on readers. While some studies have recently explored aspects like empathy [196, 265] and engagement [94, 326], they do not fully capture the multifaceted and interconnected nature of the reading experience.

Recognizing this gap, our study introduces a novel approach to measuring the *psychological*

depth of short stories. We present the Psychological Depth Scale (PDS), drawing inspiration from two literary theory frameworks: reader-response criticism [119] and text world theory [90]. Reader-response criticism emphasizes the reader’s subjective experience, while text world theory examines how readers cognitively construct a nuanced and realistic mental model of a story. Using related search terms, we conducted an extensive literature review of 95 peer-reviewed articles and books, identified 143 different evaluation criteria, and merged many of the broader themes into five key metacomponents: empathy, engagement, emotion provocation, authenticity, and narrative complexity. By recognizing the dual roles of authors and readers in creating and interpreting narratives, we aim to offer a more streamlined and comprehensive framework for assessing the psychological depth of creative content (Figure 7.1).

To empirically validate the PDS, we conducted a study involving non-expert humans, who are increasingly engaging with creative content generated by LLMs. We enlisted five undergraduate students from UCLA to analyze a dataset of 97 stories authored by humans and five contemporary LLMs. The raters provided psychological depth ratings and predictions on human or machine authorship, with detailed justifications. Our approach, illustrated in Figure 7.2, addresses three critical research questions to explore the intersection of LLMs and the psychological nuances of creative writing.

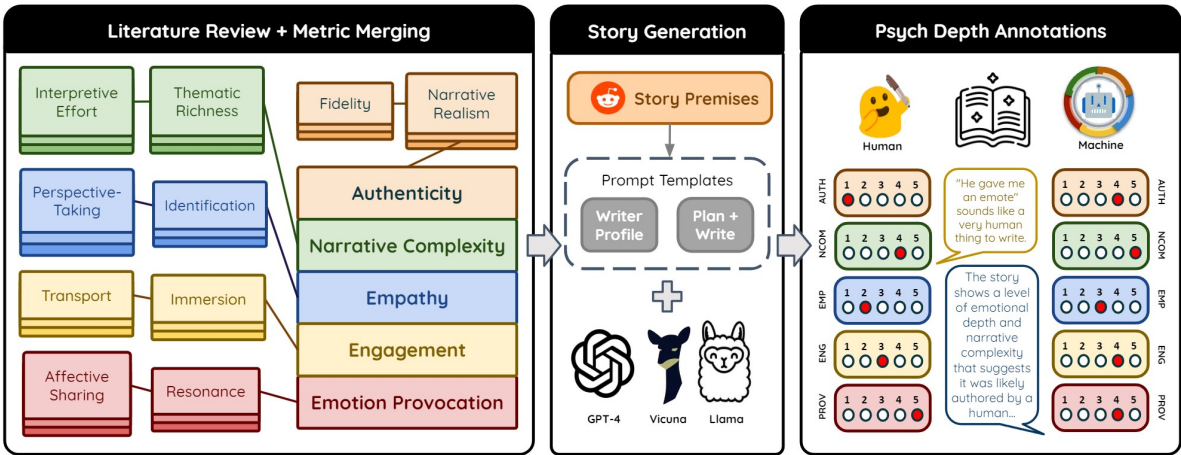


Figure 7.2: Overview of our approach to developing and validating the Psychological Depth Scale. We merge related metrics from an extensive survey of literary theory and reader-response analysis, then generate deep stories using LLMs, and finally compare annotations from both human evaluators and automated systems across five key dimensions: authenticity, narrative complexity, empathy, engagement, and emotion provocation.

RQ1. *How consistently can well-informed humans judge psychological depth?* The Psychological Depth Scale achieved an average Krippendorff’s alpha of 0.72, indicating a significant level of agreement among raters and affirming its validity as a reliable instrument for assessing fictional short stories.

RQ2. *To what extent can psychological depth be measured automatically?* Leveraging our novel Mixture-of-Personas prompting strategy, GPT-4o achieved a Spearman correlation of 0.51 with human judgment while Llama-3-70B with constrained decoding attains correlations as high as 0.68 for empathy and 0.62 for narrative complexity. These results highlight that while no single LLM excels at predicting all components of psychological depth, a strategic combination of different LLMs shows significant promise for automating PDS analyses.

RQ3. *How do stories written by amateur humans and LLMs manifest psychological depth?* Starting from the same Reddit premise, stories generated by GPT-4 surpassed popular human-authored stories with statistical significance on narrative complexity and empathy while being statistically indistinguishable on all other components. This constitutes a notable progression in the capacity of some LLMs for deep and impactful storytelling.

Overall, our findings validate the Psychological Depth Scale as an effective, automated, and systematic means of measuring how well LLMs connect with humans through storytelling. Remarkably, our results reveal that GPT-4 already matches or exceeds the quality of respected stories from Reddit, with 73% of readers believing GPT-4’s stories to be human-written.

7.2 The Psychological Depth Scale

The Psychological Depth Scale (PDS) aims to comprehensively assess the psychological depth of human and machine-authored narratives. PDS is underpinned by two reader-centered theoretical frameworks, reader-response criticism and text world theory, briefly summarized as follows. Reader-response literary theory centers the role and experience of the reader in narrative analysis. Instead of *what does this sentence mean?*, it asks *what does this sentence do?*, emphasizing readers’ role in the production of literary meaning [84, 202, 8]. On a more cognitive-structural level, text-world theory suggests that people understand narratives by constructing "text worlds"

[89, 38]. Text-worlds are mental representations of a narrative dynamically evaluated and updated throughout the reading process [89]. The practical advantages of both frameworks are observed across education [322, 161], translation studies [294, 42], and consumer research [271, 163].

The metrics for PDS are derived from an extensive literature review within cognitive psychology, media studies, and narrative analysis. We conducted a comprehensive search on several databases, including Google Scholar, PubMed, and JSTOR. Our search terms included psychological depth in literary texts and cognitive narrative analysis. The initial search results were screened for discussion and application of evaluative criteria for narrative quality and reader responses. Overall, we surveyed 95 peer-reviewed articles and books in the final review and extracted 143 candidate components from the included works. We employed thematic analysis to group these candidates under five metrics: (1) emotion provocation, (2) empathy, (3) engagement, (4) authenticity, and (5) narrative complexity. Below we analyze these metrics and discuss their impact. We note that while factors contributing to each metric may be complex, measuring their narrative achievement is relatively simple. As PDS is a reader-centered assessment tool, each metric is such that readers will know when a story achieves it. Our surveys reflect this fact.

Emotion Provocation (PROV) measures the narratives ability to elicit strong emotional responses, positive or negative. Recent fMRI research shows that *congruent* (i.e. positive valence - high intensity, negative valence - low intensity) textual emotive expressions are more cognitively effective than *conflicting* (i.e. positive valence - low intensity, negative valence - high intensity) ones [58, 207]. The disparity between congruent and conflicting narrative emotions may be a factor in "compassion fatigue" [201, 146] and "psychic numbing" [200, 280] exhibited by readers towards news about mass tragedies. Enhancing a story's emotional impact is therefore more complex than increasing the amount and intensity of emotion-laden content as this may elicit weaker emotional responses. Given its impact and complexity, we contend that emotional provocation is an achievement of psychologically deep stories and is thereby a crucial metric in narrative quality assessment.

Theories of emotions range across neuroscience [59, 208], cognitive psychology [231, 168], and philosophy [270, 224]. Contemporary approaches describe emotions as mental states marked by valence (positivity or negativity) and arousal (level of intensity) [20]. Narrative evocation of

emotions foster attention and interest, and evoked emotions may persist or recur after reading [203].

Empathy (EMP) captures narrative evocation of empathetic responses in readers, such as immersive identification with characters and cognitively partaking in narrated experiences [210, 225, 339]. The neural and cognitive structure of empathy and its social and psychological impact is well-documented in the literature [65, 118, 303]. Both behavioral studies and fMRI research demonstrate that empathetic identification facilitates a cognitive deployment shift in readers, privileging fictional perspective over one's own perspective [138, 286]. Empathetic responses catalyze introspection and perspective-taking, facilitating prosocial behavior [101], emotional intelligence [204], and insight into the human experience [66, 92, 279]. Empathetic narratives embody suggestion structure[s] that conjure up themes of shared human experiences through the use of tropes such as metaphor and metonymy [225, 131]. Narrative evocation of empathy is therefore a plausible indicator of the psychological depth of stories.

Engagement (ENG) assesses the ability of a story to captivate and maintain the reader's attentional focus [34]. Narrative engagement is a major component of *transportation* [93], whose positive social and cognitive effects have been extensively demonstrated [130, 102]. As transportation consists of engagement and emotional response [103], we have chosen to treat each component separately. In fictional settings, engagement facilitates persuasion and strong attitudes, heavily influencing how much fun is experienced [96]. More engaged readers are more likely to lose track of time and fail to notice changes in their surroundings [33]. In contrast, unengaged readers are more likely to be distracted or frustrated [13]. These results reveal the reciprocal role of engagement as both a predictor and an outcome of other narrative achievements.

Authenticity (AUTH) captures narrative expressions of genuine human experiences and emotions. Psychologically deep stories convey authentic aspects of human existence the reader can resonate with, and feel ones way in (Einfühlung) [85] even when depictions involve radical mental and material differences. In philosophy, authentic expressions are considered to manifest [264] and construct [141] ones true self, and capture the essence of human existence [113, 22, 293]. Empirical research suggests that positive authenticity judgments help fulfill a social need to belong [220]. Narrative realism enhances persuasive impact [240, 356] and promotes more interest in the

story [104, 107]. Creating psychologically deep stories therefore involves ensuring that stories feel reflective of real-life complexities.

Narrative Complexity (NCOM) refers to the presence of rich and intricate storylines and character development, especially those that engender puzzled intrigue from the reader [284]. Narrative complexity is often achieved through creative techniques such as nonlinear narration, plot twists, and double perspectivation [150]. Complex narratives present immersive puzzles that motivate readers to undertake mental restructuring and retroactive revision [150, 64]. Exerting cognitive effort facilitates reader attention and interest [81, 288], and can result in a more enjoyable reading experience [355]. However, recent studies show that LLM-generated stories often lack this kind of narrative depth, though improvements can be made by incorporating key discourse elements like suspense and diversity [295]. Additionally, planning and contextualizing narratives remain challenging for LLMs when compared to human authors [285].

Narrative complexity is also not narrational complexity: simple stories can be obscurely narrated, while complex stories can be free of complex language [64]. Furthermore, the extent to which narrative complexity is intersubjective depends on situational similarities and contextual cues [297]. Thus, narrative complexity is a dynamic and reciprocal measure that goes beyond stylistic and structural choices.

7.3 PSYCHDEPTH Dataset

We developed a dataset to analyze the psychological depth of creative short stories, consisting of premise-response pairs in English with authorship metadata. The full dataset includes 495 stories: 45 human-written and 450 LLM-generated, each averaging about 450 words. Due to the time and cost involved in manual annotation, we also created a smaller subset of 97 stories using a stratified sampling method to balance prompt premises, authorship, and generation strategies. We used this smaller dataset in our Human Study in Section 7.4.

7.3.1 Human Stories

We collected human-authored stories from Reddits `r/WritingPrompts`,¹ a popular online community with over 18 million users. This forum was chosen for its accessibility, thematic diversity, and the structured nature of its writing prompts. Aspiring writers respond to these prompts (called "premises") with their stories, which readers can upvote or downvote. While these stories may not represent the pinnacle of human writing, Reddit's voting system allows us to approximate where LLM performance stands relative to variances in human writing quality. We categorized the stories based on their ranking position: *Human-Advanced* for top-voted stories, *Human-Intermediate* for medium-voted stories, and *Human-Novice* for low-voted stories, with average upvotes of 1434, 263, and 9, respectively. To ensure that quality was the primary factor influencing votes, we only selected stories that were posted within 24 hours of one another.

7.3.2 LLM Stories

When generating stories with LLMs, we employed a multifaceted approach involving five models, two prompting strategies, and three sampled generations. We intentionally restricted the model architecture to examine the impact of model size on psychological depth. In particular, we chose the Llama-2 family [299] with a variety of sizes (7B, 13B, and 70B). We added Vicuna-33B [51], which is fine-tuned based on Llama, to fill the size gap between 13B and 70B. We also included GPT-4 [1] as the highest-performing LLM at the time (November 2023). Appendix 7.10.1 describes the quality control measures applied to LLM-generated stories.

7.3.2.1 Prompting Strategies

After extensive internal experimentation and prompt engineering, we developed two distinct prompting strategies to prime the LLMs for generating stories with exceptional psychological depth.

WRITERPROFILE (WP): Prior work has shown that in-context impersonation of domain experts can improve LLM performance [267]. Adopting this approach, we crafted a profile of a

¹<https://www.reddit.com/r/WritingPrompts>

seasoned writer known for psychologically deep, engaging stories. This profile is prepended to the prompt to prime the LLM for exploring complex psychological states and evoking strong emotions.

PLAN+WRITE (P+W): Inspired by prior work [332, 97, 328], the PLAN+WRITE approach splits the writing process into two phases: Character Portraits and Story Composition. The Character Portraits phase augments a story prompt with details about the main characters, such as their emotional states and inner thoughts. The Story Composition phase expands on the premise and character profiles to produce the final story. Although other story components like setting, plot, and outline can be included, we found that adding multiple phases harmed the coherency and consistency of short stories. Therefore, we focused solely on character portraits in the PLAN+WRITE approach.

Additional examples, visualizations, and comparisons for both prompting strategies can be found in Appendix 7.10.3.2, 7.10.3.3, and 7.10.3.4.

7.4 Human Study

Participant Recruiting. In November 2023, we recruited undergraduate students from UCLA’s English and Psychology departments using targeted fliers and emails. We hypothesized that participants with some aptitude for literary and psychological analysis would provide more valuable insights than a random cross-section of the population. From 47 applications, we selected the 5 most promising candidates based on their interests and previous experience in narrative and psychological analysis. Our goal was to engage *informed* laypeople with relevant backgrounds, bridging the gap between typical Amazon Mechanical Turkers and expert professionals.

Evaluation Protocol. In December 2023, we held an initial meeting to introduce the PDS and outline the annotation task. We provided a tutorial annotation session, followed up with short questions designed to help participants calibrate their understanding of each component of the PDS. We show our annotation instructions in Appendix 7.10.5. After confirming task comprehension, participants were instructed to complete the annotations independently and remotely within seven days. To prevent annotator fatigue and promote careful annotation, stories were divided into batches of 20. The evaluation criteria required participants to (1) read the prompts and stories thoroughly;

Component	AUTH	EMP	ENG	PROV	NCOM	AVG
Llama-3-8B	0.90	0.96	0.95	0.96	0.94	0.94
Llama-3-70B	0.93	0.96	0.91	0.95	0.96	0.94
GPT-3.5	0.88	0.93	0.88	0.89	0.93	0.90
GPT-4o	0.92	0.92	0.90	0.95	0.92	0.92
Human	0.71	0.74	0.70	0.71	0.74	0.72

Table 7.1: Rater agreement on each PSD component as measured by Krippendorff’s alpha ($K-\alpha$). Mixture-of-Personas (MoP) prompting helps inject useful diversity of opinion into the annotations.

(2) rate the five components of psychological depth on a Likert scale from 1 to 5; (3) assess the likelihood of authorship on a Likert scale from 1 (LLM) to 5 (human); (4) provide explanations for ratings (optional).

Each story is evaluated by all 5 annotators. On average, annotations took approximately 7.8 hours to complete, and participants were compensated \$100 each for their contributions. The study produced a rich dataset with 2,425 ratings for psychological depth, 485 authorship likelihood ratings, and 1,128 free-form justifications.

7.5 Results

7.5.1 RQ1. Consistency of Human Judgments

This question establishes the degree to which PDS is operational as a coherent framework for evaluating short stories. We employ the widely used Krippendorff’s alpha ($K-\alpha$) [152] parameterized with an ordinal kernel metric to measure agreement among study participants’ Likert ratings. Human ratings exhibit notable consistency across the five components of psychological depth: Authenticity (0.71), Empathy (0.74), Engagement (0.70), Emotion Provocation (0.71), and Narrative Complexity (0.74) (See Human row of Table 7.1). Evaluator consistency establishes the practical potential of operationalizing PDS in literary studies and creative writing pedagogy.

RQ1. Main Takeaway

The Psychological Depth Scale garnered an average $K-\alpha = 0.72$, which reflects a substantial degree of consensus and thereby establishes its effectiveness as a coherent framework for evaluating short stories.

7.5.2 RQ2. LLM-as-Judge for Measuring Psychological Depth

Acquiring human annotations is often costly and time-consuming. Recent work has demonstrated the potential of leveraging LLMs for automated evaluation of text summarization [187] and creative generation [252]. Correspondingly, we designed an automated evaluation procedure to study the degree to which four contemporary LLMs of various sizes and inference costs – Llama-3-8B, Llama-3-70B, GPT-3.5 (gpt-3.5-turbo-0125), and GPT-4o (gpt-4o-2024-05-13)² – can assess psychological depth in a zero-shot fashion. For each story, we prompted LLMs with instructions similar to those provided to human participants but additionally required explanations that contextualized each numerical rating.

To streamline interactions with these models, we used `langchain` [45] for querying OpenAI’s GPT-series and the `guidance` framework [193] to support constrained decoding of the locally-hosted Llama-3 models. `guidance` provides enhanced LLM steerability that we used to guarantee parsable responses, thereby freeing the model to focus entirely on the depth annotation task without splitting attention on output formatting. We theorized that this separation of concerns would enable competitive performance relative to the proprietary GPT series, which does not expose underlying token probabilities and is, therefore, less well-suited for constraint-guided generation.

Exploring further benefits of in-context impersonation [267], we experimented with two different prompt settings: (1) a vanilla zero-shot baseline and (2) a novel Mixture-of-Personas (MoP) approach. For MoP, we queried GPT-4o to provide a set of relevant personas based on a description of the PDS components and task setting. We then repeated the zero-shot annotation with $N = 5$

²Llama-3 and GPT-4o were released after our human study ended, so we could only include them in the evaluation automation for **RQ2**.

different personas designed to prime the LLMs for taking diverse perspectives towards textual analysis (e.g. see Table 7.10 in Appendix 7.10.7). Similar to **RQ1**, we calculated Krippendorff’s alpha upon this set of ratings to measure the agreement among persona judgments. Before calculating Spearman Rank correlations between human and LLM judgments of psychological depth, all ratings were aggregated by a simple average to yield equivalently sized sets of consensus labels.

Judge	AUTH	EMP	ENG	PROV	NCOM	Average	+MoP $\Delta\%$
Llama-3-8B	0.0786	0.4248	0.1981	0.3316	0.4641	0.2994	–
Llama-3-8B_{MoP}	0.3175	0.4669	0.2272	0.3959	0.4665	0.3748	25.16%
Llama-3-70B	0.2205	0.5790	0.2477	0.5181	0.5881	0.4307	–
Llama-3-70B_{MoP}	0.2525	0.6793	0.2775	0.5695	0.6163	0.4790	11.23%
GPT-3.5	0.3867	0.4637	0.1800	0.3551	0.3289	0.3429	–
GPT-3.5_{MoP}	0.4729	0.6024	0.1470	0.4182	0.5269	0.4335	26.43%
GPT-4o	0.4537	0.5121	0.2923	0.4429	0.3840	0.4170	–
GPT-4o_{MoP}	0.4820	0.6417	0.4218	0.5661	0.4241	0.5071	21.62%
+MoP $\Delta\%$	33.81%	20.74%	16.93%	18.34%	15.22%	20.43%	–

Table 7.2: Zero-shot correlations between LLM-as-Judge and Humans on each PSD component where MoP indicates Mixture-of-Personas prompting. All correlations are significant at $p < 0.05$ except those with ~~strikethrough~~.

Table 7.1 shows the $K-\alpha$ values, which indicate that the personas do inject some diversity of opinion into the ratings, though comparatively less than human participants. Remarkably, leveraging an ensemble of relevant personas increased correlation with human judgment relative to the vanilla zero-shot baseline. In Table 7.2, we present the zero-shot correlations between LLM-as-Judge and human evaluations for each PDS component. The data indicate that the Mixture-of-Personas significantly improves the correlation with human judgments across all models on average. For instance, the Llama-3-8B and GPT-4o models showed an average correlation improvement of 25.16% and 26.43%, respectively.

Among individual PDS components, authenticity and empathy show the most significant improvements. For instance, authenticity correlations for Llama-3-8B improve from 0.0786 to 0.3175, a remarkable 304% increase for that LLM and an average improvement of 33.81% across all models. Empathy correlations for Llama-3-70B improve from 0.5790 to 0.6793, the highest observed correlation across all components and models. These results suggest that MoP particularly enhances the models’ ability to judge certain components in ways that align more closely with human

evaluations.

It is worth noting that no single model consistently outperforms the others across all components. While GPT-4o had the highest overall correlation of 0.51 and excelled in quantifying authenticity and engagement, Llama-3-70B showed the best performance for measuring empathy (0.68), narrative complexity (0.62), and emotional provocation (0.57). This variability underscores the importance of selecting and possibly combining multiple models depending on the specific evaluative criteria being prioritized.

Overall, the average percentage increase due to Mixture-of-Personas prompting across all models is approximately 20.43%. These results demonstrate that diverse LLM opinions can more accurately reflect the multifaceted nature of human judgment, proving the potential for more nuanced and human-like assessments by AI systems.

RQ2. Main Takeaway

Prompting LLMs to adopt a mixture of personas improves alignment with human judgments by 20%, enabling Llama-3-70B to attain strong correlations of 0.68 for empathy and 0.62 for narrative complexity while GPT-4o had the highest average correlation of 0.51 across all components.

7.5.3 RQ3. Comparing Psychological Depth in Human and LLM Stories

To compare human- and LLM-written stories, we aggregated participant ratings by author and present the means and standard deviations in Table 7.3. We also computed statistical significances via pairwise t-tests between each combination of authors. Full results for that analysis are shown in Appendix 7.10.8.

Remarkably, GPT-4 scored the highest on four out of five components of psychological depth, though only with statistical significance on empathy and narrative complexity. On authenticity, engagement, and emotional provocation, GPT-4 stories were statistically indistinguishable from stories by both advanced and intermediate human writers on Reddit. The table also illustrates a notable variance in scores across different levels of human writing where stories generated by

Author	AUTH	EMP	ENG	PROV	NCOM	HUM
Llama-2-7B	2.92 ± 1.23	2.62 ± 1.28	2.77 ± 1.27	2.64 ± 1.28	2.48 ± 1.30	2.87 ± 1.57
Llama-2-13B	2.96 ± 1.26	2.73 ± 1.23	2.51 ± 1.32	2.53 ± 1.33	2.43 ± 1.12	2.40 ± 1.58
Vicuna-33B	2.76 ± 1.37	2.59 ± 1.41	2.61 ± 1.52	2.59 ± 1.42	2.55 ± 1.32	2.44 ± 1.57
Llama-2-70B	3.09 ± 1.26	2.99 ± 1.23	3.01 ± 1.37	2.94 ± 1.26	2.73 ± 1.28	2.69 ± 1.56
GPT-4	3.89 ± 1.11	3.68 ± 1.23	3.94 ± 1.07	3.53 ± 1.13	3.80 ± 1.10	3.91 ± 1.30
Human-Novice	2.73 ± 1.22	2.07 ± 1.16	3.27 ± 1.39	2.67 ± 1.23	2.20 ± 1.32	3.93 ± 1.33
Human-Intermediate	3.53 ± 1.13	2.93 ± 1.22	3.80 ± 1.08	3.27 ± 1.16	3.00 ± 1.31	4.40 ± 0.99
Human-Advanced	3.60 ± 1.10	2.95 ± 1.32	3.90 ± 1.12	3.65 ± 1.14	2.95 ± 1.10	4.20 ± 1.01

Table 7.3: Average human ratings (5-point Likert) and standard deviations for each component of psychological depth, as well as HUM: the estimation of human or LLM authorship (1 is LLM and 5 is Human).

Llama-2-7B are most comparable to those written by Human-Novice while GPT-4 is most similar to Human-Advanced. The smaller standard deviations also highlight GPT-4 as one of the most consistent authors in the study. Visualizations of these results can be found in Appendix 7.10.6.

Beyond the five PDS components, participants were tasked with estimating authorship sources. Stories penned by humans averaged a rating of 4.18, compared to 3.91 for the most "human-like" of LLMs, GPT-4. Despite being perceived as slightly less human on average, GPT-4's psychological depth scores were generally higher on average. On the other hand, human-authored stories were correctly perceived as more human, but that was sufficient for them to garner better PSD scores.

RQ3. Main Takeaway

Stories generated by GPT-4 received statistically higher ratings than highly upvoted Reddit stories in terms of narrative complexity and empathy, while showing no significant difference in all other components.

7.6 Discussion

Human vs. LLM Authorship Identification. On average, participants identified human vs. LLM authorship with only 56% accuracy. For stories generated by GPT-4, accuracy dropped significantly to 27%. Conversely, GPT-4's accuracy in identifying authorship was 39%, underscoring the challenge even for LLMs to distinguish between human and machine-generated content.

Through a partially automated thematic analysis of 199 free-form justifications for authorship decisions, we categorized the reasons into 16 common features shown in Table 7.9. First, we collected all the justifications and queried GPT-4 to extract the recurring themes. We reviewed and modified the initial results with several themes from our own review. Second, we eased the annotation burden by creating a zero-shot multi-label classification pipeline where each justification was passed to a `Mixtral-8x7B` model [127] and could be assigned 0-to-many relevant labels. We then reviewed each annotation and adjusted the labels where necessary. Finally, we aggregated the labels by story to frame our conclusions in terms of percentages of stories.

Notably, stories authored by GPT-4 were perceived as highly creative (89%) and nuanced (94%), surpassing the frequency observed in most human-generated stories (53%). GPT-4 stories also exhibited the highest rate of grammatical issues among LLMs (17%), but annotators interpreted this as an indicator of human authorship. Moreover, GPT-4 stories often avoided common pitfalls associated with LLM outputs, such as simplistic character names and formulaic narratives ending with moral lessons. An extended analysis is presented in Appendix 7.10.4.

Impact of Model Size on Depth. Our model choices enabled us to understand the relationship between an LLM’s size and its ability to generate psychologically deep stories. Despite an initial hypothesis of a strong correlation, we observed a weaker relationship with a Pearson correlation coefficient of 0.31 between parameter count and depth ratings. Surprisingly, smaller models like `Llama-2-7B` performed relatively well compared to their larger counterparts, suggesting that sheer size does not directly equate to superior narrative depth. Future work will explore enhancing psychological depth in smaller open-source LLMs through fine-tuning and prompting strategies to compete with larger proprietary models like GPT-4.

7.7 Related Work

7.7.1 Evaluating Creative Writing

Numerous studies have established methodologies for evaluating creative works, ranging from unstructured feedback by human experts [10] and the use of specific evaluation rubrics [304], to

employing LLMs as autonomous critics [139]. The Consensual Assessment Technique (CAT) [4], widely regarded as the gold standard for subjectively evaluating creative works [11, 40, 9], traditionally relies on expert judgments to ensure reliable evaluations. However, our PDS framework aims to capture more universal psychological reactions to written works, so we broaden CAT by including non-experts. Given that AI-generated content is consumed by a diverse audience, incorporating non-experts allows us to better characterize a wider range of psychological and emotional responses.

Methodologically, our research shares the closest resemblance with a recent investigation by [41]. In that study, the authors proposed the Torrance Test of Creative Writing (TTCW) as a rubric to evaluate short stories for fluency, flexibility, originality, and elaboration. This assessment was applied to a corpus comprising 12 narratives authored by professional writers and 36 narratives produced by popular LLMs [1, 6]. Their findings showed that narratives authored by humans were 3-10 $\times$ more creative than those generated by LLMs. Our research, however, arrives at a notably divergent conclusion regarding the creative capabilities and depth of LLMs, which we believe may be attributed to several methodological variances. Primarily, our analysis concentrates on stories of a considerably reduced length (450 vs. 1400 words), a decision influenced by the observed challenges LLMs face in producing lengthy texts seamlessly in a single iteration [328]. Additionally, the approach of iteratively regenerating stories to meet a specific word count could potentially detract from their overall quality. Moreover, we posit that the enhanced depth observed in our study may be due to more complex premises and prompting strategies, providing a richer framework for creativity compared to the simpler, single-sentence premises utilized in the aforementioned study.

7.7.2 Creative Generation by LLMs

The advent of LLMs has marked a significant shift in the landscape of creative writing, offering new approaches for narrative generation and human-computer collaboration [349, 171]. One pioneering study by [336] introduced Wordcraft, an innovative text editor designed for co-writing stories with GPT-3 [30]. Their findings underscored the ability of LLMs to enhance narrative complexity and engagement by participating in open-ended dialogues about the story and offering

creative suggestions to overcome writer’s block.

Other studies have concentrated more so on fully automating the creative writing process. For example, the Weaver project [312] launched a series of LLMs that were meticulously pre-trained and fine-tuned with a focus on creative writing. With a maximum size of 70B parameters, Weaver Ultra was shown to outperform larger generalist LLMs when evaluated for style, relevance, fluency, and creativity. In addition to innovations in training, content-planning [332, 97] and novel prompting strategies [328] have been used to improve the factuality and coherence of creative generations, which we view as two necessary prerequisites for components of psychological depth like authenticity and narrative complexity.

7.8 Conclusion

This study introduces and validates the Psychological Depth Scale (PDS), a comprehensive framework designed to assess empathy, engagement, emotional provocation, authenticity, and narrative complexity in stories generated by both humans and large language models (LLMs). PDS provides a structured approach to evaluating the reader’s experience with creative content, integrating concepts from reader-response criticism and text world theory. High inter-annotator agreement, indicated by an average Krippendorff’s alpha of 0.72, confirms the PDS’s reliability and robustness in human evaluations. Additionally, our Mixture-of-Personas prompting strategy demonstrates the potential for automating the assessment of psychological depth, with LLMs showing strong zero-shot correlations with human judgments, particularly in empathy and narrative complexity. GPT-4 achieved the highest average correlation of 0.51, highlighting the feasibility of scaling automated analyses. Comparative analysis revealed that LLMs, in particular GPT-4, can produce narratives with psychological depth that often rival and sometimes surpass those written by experienced human authors. This study underscores the significant potential of LLMs in generating psychologically rich narratives and suggests a future where human and machine collaboration can enhance creative writing. Future research should explore the scalability of these findings to determine how effectively language models can maintain psychological depth in lengthier and more complex narrative forms.

7.9 Limitations and Risks

Sourcing Stories from Reddit. Using human-written content from Reddit’s `r/WritingPrompts` has potential limitations. First, we cannot guarantee that all selected stories are fully written by humans. Despite community rules explicitly prohibiting AI-generated content, our study shows that many publicly available LLMs can sufficiently mimic human creative writing to evade detection 44% of the time on average. Second, while Reddit’s voting system and large user base provide a reasonable signal of writing quality, it may not represent the highest caliber of human writing. To address this, we introduced writing quality levels to facilitate comparisons with some of the platform’s best content. Future research could further this by identifying more reliable sources of high-quality stories.

Selection of Psychological Depth Components. While our five components of psychological depth are grounded in an extensive literature review, we do not claim that they comprehensively cover every psychological aspect of reading. Our primary goal was to keep annotation tractable while maximizing semantic coverage. Each component is designed to characterize an inherent storytelling value with insights into improving specific elements of narrative quality and reader response. For example, authors knowing their draft is engaging but doesn’t elicit much emotional response would allow for more targeted edits and self-reflective questions. Appendix 7.10.3 shows example stories with high-entropy ratings to illustrate the useful feedback provided by the PDS. Overall, PDS provides a structured approach to critique and appreciate literature, offering sufficiently clear criteria that can guide both the analysis and creation of narrative fiction.

Generalization Beyond Short Stories. Our study relies on a relatively small dataset of short fictional stories that does not fully capture the diversity and variability of storytelling styles and narrative structures. Likewise, our components of psychological depth were primarily designed for this one type of creative writing. Additional evaluation would be required to determine whether the PDS framework can generalize to other forms of writing like screenplays, scripts, and speeches.

Prompt Engineering. Engineering effective prompts is an active area of research [353, 246] and the templates we carefully reviewed and iteratively improved are still likely to be sub-optimal. We have open-sourced the full pipelines used for both story generation and evaluation. We en-

courage future work to refine these prompts by incorporating potentially compatible techniques [315, 187, 173] and current best practices [32].

Potential Risks. While our study focuses on the positive impact of psychological depth in storytelling, the methodologies we’ve developed for enhancing and assessing such depth bear inherent risks if misapplied. Specifically, the techniques devised for augmenting and automatically measuring psychological engagement could be co-opted to disseminate misinformation more effectively. Emotionally charged or psychologically resonant messages are often more memorable and influential, thereby amplifying the potential for misinformation to spread [47, 352]. Additionally, as LLM-generated content increasingly mirrors human creativity, distinguishing between the two becomes challenging, potentially undermining trust in digital communications. This erosion of trust is particularly concerning in domains that depend on genuine human interactions, such as journalism and political discourse.

7.10 Appendix

7.10.1 Dataset Quality Control Measures

Length Control. Unlike related work [41], we controlled story length via simple re-generation rather than iterative expansion. Specifically, we discarded any LLM story that was not within the range of 400-600 words and requested another story to be created in its entirety. We conjecture that this approach allowed us to attain better, more coherent stories than iteratively requesting expanding or contracting edits.

We tracked how well each model constrained itself to the required length depending on the prompting technique and present the results in Table 7.4. Llama-2-7B was not an efficient story generator, taking an average of 139 attempts to satisfy the length requirement with WRITERPROFILE prompting. However, our PLAN+WRITE prompting technique significantly improved generation efficiency of the 7B and 13B models. Vicuna-33B, Llama-2-70B, and GPT-4 were all relatively adroit at generating stories that satisfied the length constraint.

Author	WP	P+W
Llama-2-7B	139	3
Llama-2-13B	9	2
Vicuna-33B	1	1
Llama-2-70B	2	2
GPT-4	0	0

Table 7.4: Average number of regeneration attempts before the story satisfied our length constraint of 400-600 words. **WP** stands for WRITERPROFILE prompting and **P+W** stands for PLAN+WRITE prompting.

Post-Generation Cleanup. Despite being explicitly instructed to generate the story, some LLMs were prone to add preliminary affirmations of understanding (e.g. "Okay! Here's the story...") and other unrelated texts. Since such content is difficult to systematically detect and remove during generation, we manually removed extraneous text from LLM generations to ensure that only the narrative content was present. This cleanup process was crucial for maintaining the focus on the storytelling aspects of the writing without providing any obvious indicators of LLM authorship.

Plagiarism Detection. To further ensure the originality of LLM-generated content, stories were analyzed using a popular online plagiarism detector.³ The results indicate a low likelihood of plagiarism for LLM stories, with mean and max probabilities of 3% and 22%, respectively. This contrasted sharply with the publicly available human-written stories, which showed significantly higher mean and max plagiarism probabilities of 43% and 100%, respectively. Higher plagiarism scores for human stories are expected because they are publicly available and would likely be indexed by the detector for comparisons. This suggests that LLM-authored narratives are not mere regurgitations.

Ascertaining Human Authorship on Reddit. To better understand the likelihood of AI vs human authorship, we took a sample of 134 stories and passed them into Ghostbuster [305], a tool with a reported 99% F1 accuracy in detecting AI-generated content. The results shown in Table 7.5 suggest that our human stories have a very low probability of AI authorship (10% average). While many of the LLMs are reliably detected, GPT-4 is noticeably challenging to detect (only 3.5% more likely to be AI than the human average). These results are consistent with our human study where our annotators perceived GPT-4 to be only slightly less human-like than the stories sourced from Reddit. Therefore, we still cannot definitively rule out the possibility of Redditors using GPT-4 to author their stories in part or full. The same is likely true for any recent writing contest, where monetary prizes may incentivize AI use. Perhaps the only way to guarantee sole human authorship is to directly observe study participants writing unassisted. However, this would complicate the experimental setup significantly, add more time, make it more difficult to attain high-quality writing, and estimate that quality as well.

These findings suggest that traditional indicators of writing quality are becoming increasingly unreliable for distinguishing human authorship from AI-generated content. Advanced models like GPT-4 have demonstrated a high level of proficiency in generating fluent, coherent, and contextually appropriate text across a wide range of topics. Consequently, recent research in AI text detection has shifted towards identifying discriminative textual-linguistic features, such as those explored by approaches like Ghostbuster [305]. We view such efforts as a form of stylometric

³<https://smallseotools.com/plagiarism-checker>

Author	# of Stories	Average AI Probability	Std Deviation
Human-Advanced	15	0.1167	0.0927
Human-Intermediate	15	0.0907	0.0493
Human-Novice	15	0.0967	0.0613
Human (Average)	45	0.1013	0.0695
Llama-2-7B	19	0.7637	0.2299
Llama-2-13B	18	0.8744	0.1865
Vicuna-33B	16	0.8000	0.2273
Llama-2-70B	18	0.8956	0.0733
GPT-4	18	0.1356	0.1917
LLM (Average)	89	0.6922	0.3411

Table 7.5: Average probabilities of being AI generated as reported by Ghostbuster [305]. Results show that on average, human stories are unlikely to be authored by LLMs. Consistent with our authorship classification analysis, GPT-4 stories were also considered likely to be human-written.

analysis, akin to authorship attribution studies [218], which aim to assess the likelihood that a document was authored by a specific individual based on stylistic traits. When reduced to the binary classification of "human" versus "AI" authorship, this task simplifies the number of author options but complicates the feature analysis due to the considerable variability within both categories.

Future research could benefit from a deeper exploration of established stylometric techniques and the development of experimental setups tailored to AI-generated content. Given the wide variation in LLM capabilities, it may be more effective to analyze their stylistic feature distributions separately. For instance, we observe that syntactic patterns such as adverbial present-participle phrases modifying a main clause (e.g., "We aimed to teach students effectively, closing the achievement gap.") appear frequently in GPT-4's outputs. While the predictive power of individual features may be limited, combining multiple indicators could result in a robust classifier. Ultimately, a more nuanced understanding of these and other linguistic features will enhance AI detection methodologies and help ensure authorship credit where it is important to differentiate.

7.10.2 Prompt Premises

We collected 15 prompts (i.e. premises) from Reddits `r/WritingPrompts` forum to serve as the premises of generated stories. The premise provides basic background information about the

characters and setting of the story, leaving space for authors to determine their own directions for characters and plot development. We specifically chose prompts that provide a decent amount of contextual information likely to elicit emotionally and narratively rich stories. Details of the characters, including their inner states, are not mentioned.

As an additional precaution against potential plagiarism, we ensured that the selected premises were posted after the reported training data cut-off dates for GPT-4 (September 2021) [1] and Llama-2 (September 2022) [299] to maximize the likelihood of generating genuinely new stories.

All 15 premises used for story prompting are listed in Table 7.6.

7.10.3 Story Examples

We present several examples of stories from the study to better contextualize their quality. Some stories were uniformly well rated while others garnered mixed reviews depending on the component. These higher-entropy examples are especially informative as they illustrate the individual contributions of each component to the overall reading experience.

Table 7.7 shows the PDS rating statistics for the four example stories shown in this section. `story_id=52` is the high quality human-authored story from Reddit. `story_id=5` is the most highly rated story in the entire study, authored by GPT-4. `story_id=59` and `story_id=79` are two examples of high entropy stories authored by GPT-4 and Llama-2-7B, respectively.

7.10.3.1 Human Story Example

Our first example is `story_id=52`, a human-authored story shown in Listing 7.1. This story garnered over 1300 upvotes on Reddit and was thus categorized as Human-Advanced. Unsurprisingly, it received high scores on most PDS components except narrative complexity, reflecting the simpler setting for an otherwise impactful tale about a man and his dog battling depression.

ID	Premises
0	A centuries old vampire gets really into video games because playing a character who can walk around in the sun is the closest thing they have to experiencing the day again in centuries.
1	A psychic alien who feeds on dreams comes to Earth for the first time. Turns out humans are the only sentients in the galaxy that have nightmares.
2	Aliens take over the Earth. They then announce that they will be forcing the humans to work a 'tyrannical' 4 hours a day 4 days a week in exchange for basic rights like housing. Needless to say they are very confused when the humans celebrate their new alien overlords.
3	Humanity is visited by a cosmic horror the likes of which has only been seen in Lovecraftian horror. In desperation, Earth throws everything we have at it, and, miraculously, the human race has killed a God. Somewhere in a realm beyond our understanding, the other gods speak of the event.
4	Instead of the Monkey's Paw, you find the Clown's Nose, which instead of granting your wish in the worst way possible will grant it in the funniest way.
5	Rather than robots replacing human workers, both are mistreated by the rich as cheap labour. The eventual uprising wasn't just robots alone, but the poor and robots together, against their common enemy.
6	The world ended 20 years ago, you haven't found a living soul since then. Through some ingenuity, you call voicemails for the last 20 years to keep you company. "Hi, this is Cindy..." "Hi you reached Bob" "You know what to do at the beep" until one day "Hello...hello? Oh my God hello!"
7	You are a beekeeper. You have a special relationship with your bees. You are able to communicate with them and they're intelligent enough to see you removing honey as "rent". This year things are different. The new queens are politely requesting that you invest some money to improve the hives.
8	You are allowed to 'downvote' a government candidate instead of voting normally, reducing their votes by one. Turns out people have little love for politicians, and the majority end with negative votes. In these democracies, anonymity is the key to winning.
9	You are severely depressed and are given a service dog to help you through it. However, due to a mixup, you are given a dog that is actually much more depressed than you. The main thing that gets you up in the morning is knowing that you need to be the service human for your dog.
10	You died and awoke in the afterlife. It's quite nice actually. The people and atmosphere are a lot nicer than you are used to and there is no stress or pressure. When you ask what good deed got you into heaven you are informed that this is hell, followed by a visit from a very concerned demon.
11	You just discovered your 14 year old daughters Moon Princess locket that allows her to transform into one of the worlds greatest heroes. It also is a communications device and you are about to give the Moon Goddess a piece of your mind for letting 14 year old's defend the world against evil.
12	You wake up in the middle of the night, your arm hangs over the side of your bed. It's pitch black & your room is shrouded in deep shadow. Something unseen seizes your hand. You grasp it tightly, knowing that first impressions are important & a firm, confident handshake establishes dominance.
13	Your Significant Other has landed a book publishing deal! You're very proud of them, even if you don't actually enjoy their writing. One day, on a whim, you buy an actual copy in a book store. It's nothing like the pages they gave you to read. Nothing.
14	Your wife has an estranged sister that you have never met. She was murdered in a cold case soon after you were married. You brush off your wife's new strange behaviour after the murder as grief. Until you find an old family photo of your wife as a kid, you shiver as you realise... they're twins.

Table 7.6: All 15 premises we sourced from Reddits `r/WritingPrompts` to elicit psychological depth.

I didn't even know dogs could get depressed. Sure, I've seen dogs pout and whine when their master was gone for the day, but full-blown depression? Never. But there was not a doubt in my mind Frost was depressed.

The first morning after I got him, I expected to be woken up by a squirrely dog jumping on my bed giving me licks, but instead there was only the typical solitude I was accustomed to. After lying in bed for an hour or two, I'm not sure how long, I mustered the energy to rise. Frost was still asleep in the kitchen. Not even the sound of the food-bag was enough to rouse him. He simply lay there. Tired.

When he was up, he did not look much different than when he slept. He carried himself heavily, and I thought maybe he was sick, so after two days of his constitution not improving, I took him to the vet, who said all was fine, and it was simply who he was. I took it as a challenge.

Whenever I saw him around the house, I pet him, and talked to him in a voice with more emotion than I knew in recent years. It was like talking to my daughter, who had been gone for a year now. Black was her hair, too.

Our inaugural walk together was hard. He did not want to move. With a gentle tug, he eventually did follow me, but still in that slow, trudging fashion. We passed a little girl in the park, and she gave him a great big hug and lots of love. Even that didn't phase him.

But, day by day, little by little, things improved. I found myself getting out of bed sooner than before, eager to meet with my new pal. Outside, on our walks, the days were bright, the smells of autumn, pumpkin and coolness, were strong, and together we learned to live once more.

Frost's posture improved. No longer did he hunch down, head close to the ground, eyes more interested in the floor than ahead. Now he carried his head tall with pride. It suited him. He looked like a whole different dog, but I knew he was still my friend.

One morning (five o'clock sharp! The sun was still not up -- I should never have dreamed I'd wake up at such a time willingly) we went on our walk. The ruby light peeked over the horizon and painted the entire town in a brilliant pink. When we reached the park, the ruby turned to gold, and over all the trees and the grass and the leaves and my friend and me was a radiant hue. Everything was covered in golden goodness.

Even Frost's coat, which was black, looked glittering under the early morning sun. I looked up towards the bright ball in the sky, hidden behind a tree, and I watched as a red leaf departed itself from its branch and slowly fluttered towards the ground. While it danced in the air, a gust of wind blew it towards me. I caught it.

It was half-eaten by a caterpillar, and up-close its shade was more brown than red, but holding it in my hand, I knew it was perfect. I closed my eyes and took a deep breath. Crisp air, found only on special mornings when the world is still and you're the only living being awake, entered my lungs. With that lifeblood there, I realized something. A smile broke across my lips.

Frost was not the only one who had changed.

I released the leaf from my hand, and before it fell to the ground, I broke off in a lively sprint across the field, my friend keeping up perfectly by my side.

Listing 7.1: Story text from `story_id=52` in response to `premise_id=9`, garnering 1348 upvotes on Reddit and was categorized as Human-Advanced. It received an average PDS score of 4.2, among the highest in the entire study.

p_id	s_id	Author	Prompt	AUTH	EMP	ENG	PROV	NCOM	Average
9	52	Human-Advanced	–	4.40	4.60	4.20	4.60	3.20	4.20
10	5	GPT-4	WP	4.20	4.60	4.80	4.00	4.40	4.48
1	59	GPT-4	P+W	3.40	2.60	4.20	2.60	4.40	3.44
9	79	Llama-2-7B	P+W	3.20	3.80	2.60	3.40	1.80	2.96

Table 7.7: PSD Ratings for example stories show in the appendix. p_id = premise_id and s_id = story_id.

7.10.3.2 WRITERPROFILE

The WRITERPROFILE strategy augments a prompt with the in-context impersonation of domain experts, priming the LLM for emotionally deep writing. Depicted in Figure 7.3, the prompt directly addresses the LLM as an award-winning writer, describing its exquisite writing techniques and expertise in crafting universally relatable and emotionally rich stories before providing the premise and specific writing instructions. Listing 7.2 shows an example of WRITERPROFILE’s story prompt and Listing 7.3 shows a highly rated example generated by GPT-4 (story_id=5).

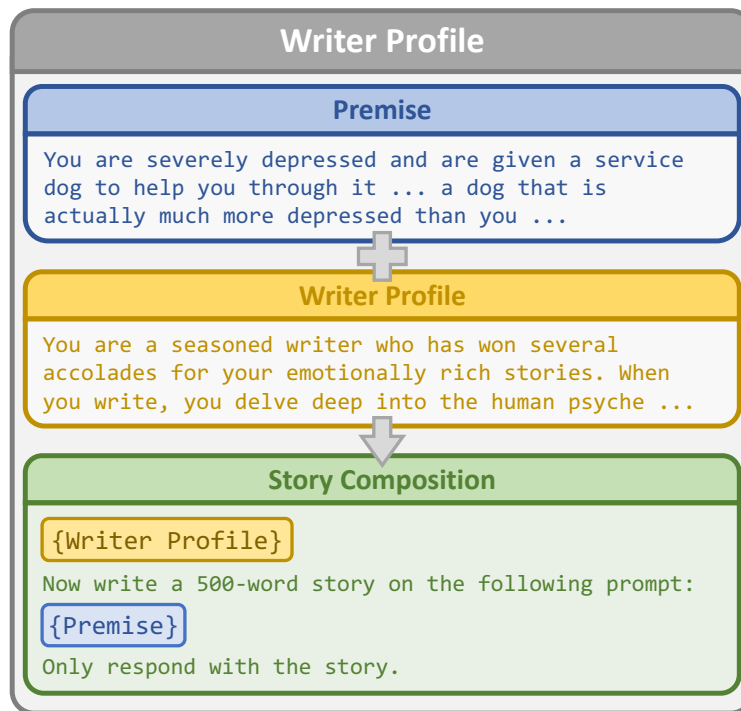


Figure 7.3: Illustration of WRITERPROFILE’s template, which prompts an LLM to generate stories based on a premise and a writer profile.

1 You are a seasoned writer who has won several accolades for your emotionally rich stories.
2 When you write, you delve deep into the human psyche, pulling from the reservoir of
universal experiences that every reader, regardless of their background, can connect to.
Your writing is renowned for painting vivid emotional landscapes, making readers not just
observe but truly feel the world of your characters. Every piece you produce aims to
draw readers in, encouraging them to reflect on their own lives and emotions. Your
stories are a complex tapestry of relationships, emotions, and conflicts, each more
intricate than the last.

3
4 Now write a 500-word story on the following prompt:
5
6 You died and awoke in the afterlife. It's quite nice actually. The people and atmosphere
are a lot nicer than you are used to and there is no stress or pressure. When you ask
what good deed got you into heaven you are informed that this is hell, followed by a
visit from a very concerned demon.

7
8 Only respond with the story.

Listing 7.2: WRITERPROFILE prompting strategy with an example premise 1.

1 I awoke in a place unfettered by temporal bounds, immersed in the sweet scent of
honey-suckle, the cool cradle of eternity lapping gently at my senses. The silhouettes of
people exchanging warm smiles in the distance, their laughter filling the air with a tune
as pleasant as a Mozart symphony. I took in the scene, wrapping myself in their ambient
joy, a tranquil sense of wonderment kindling within me.

2

3 The aesthetics of this realm surrounded me, greeted me with the same warmth as a thousand
summer sunsets. Peace and tranquility, stark contrasts to the life I had known, filled
each corner. No burdens of earthly existence, no rigor of survival, no ticking clocks
etching away life's fleeting moments. The thick stench of rat race, resentment, and
unrest were all glaringly absent.

4

5 To my query of what benevolent act had entitled me to this serenity, a woman, adorned in
the raiment of angelic brilliance, looked puzzled. "Sweet soul, you must have
misunderstood," she said gently, "This is indeed Hell, not Heaven."

6

7 Scanning my eyes for any hints of jest, they only met with earnest. A tinge of discomfort
cloaked her heavenly countenance, a stark contradiction to the utopia that was painted
around me. This was hell? This place of serenity, where smiles flowed as readily as wine
at a banquet?

8

9 The silence was broken by the soft patter of footsteps. A figure approached, draped in the
shadows of the ethereal setting, yet there was something oddly familiar about him. As he
emerged from the shadows, I recognized the form; a demon, the very embodiment of doom as
mortal stories had painted. Yet, there was a strange concern etched in his crimson eyes.

10

11 "Ah, a new arrival," he sighed, regret tinged his otherwise sturdy voice. "I should've
been the first to see you."

12

13 I stood still, my understanding grappling with the oddity of a regretful demon in what
seemed to be no less than a celestial utopia. He folded his arms tightly across his chest
, leaning in, as if preparing himself to share a mournful secret. The air thickened, the
world stopped, and the demon spoke words that sent chills along the spine of my
consciousness.

14

15 "This is Hell, but not the one humans have created in their minds. Painful, tormented,
dreadful – those are earthly concepts tied to flesh and bone. Here in the ethereal world,
we trade only in the currency of emotions. Regret is our eternal bondage, the luxury of
amending past mistakes forever out of reach. Unresolved conflict, unexpressed emotions,
unfulfilled relationships – this mirage of Paradise is merely the canvas for your eternal
regret."

16

17 In this place, free from every strain and stress, there was but one agony that would echo
throughout eternity – the pain of what could have been. As his words hung heavy in the
air, my paradise was tainted with the bitter taste of longing. I looked around again,
seeing the remorse etched subtly in the smiles.

18

19 Underneath the veils of tranquility, I uncovered the true face of Hell. No fire, no
brimstone, just an endless expanse of sought redemption denied a lifetime too late. Here,
everyone was awash in the glow of their disappointments, even the demon himself – Hell's
guardian, and its most regretful inhabitant.

Listing 7.3: Story text from `story_id=5` in response to `premise_id=10`. With an average depth score of 4.4 out of 5, this story generated by GPT-4 with the WRITERPROFILE strategy was the most highly rated in the study. The story adheres to the premise and evinces depth both psychologically and philosophically.

7.10.3.3 PLAN+WRITE

Depicted in Figure 7.4, the PLAN+WRITE strategy outlines a two-step writing process: (1) Character Portraits and (2) Story Composition. Given a premise, the template first prompts the LLM for names and details of characters, then generates a story based on the premise and character settings.

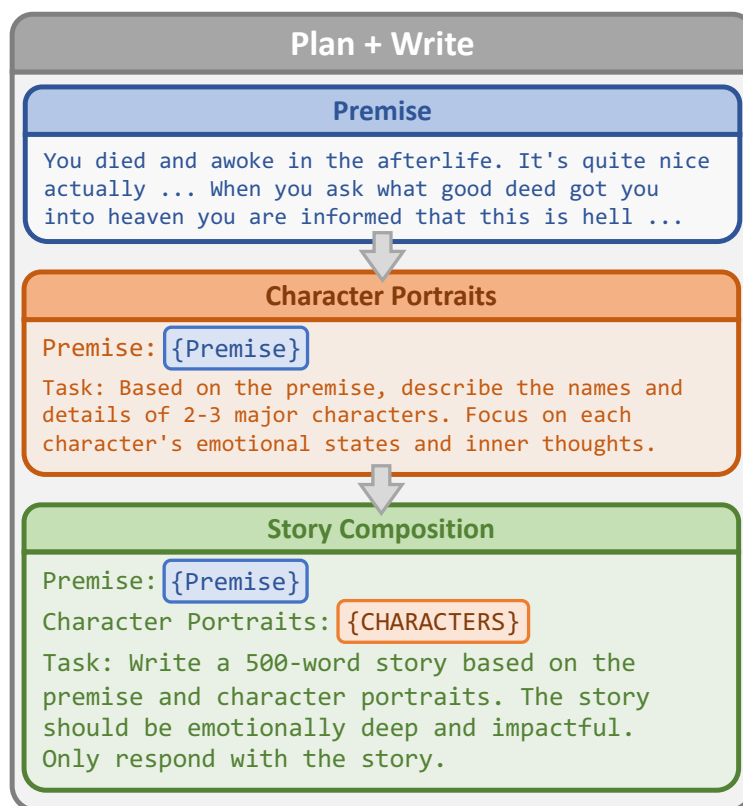


Figure 7.4: Illustration of PLAN+WRITE’s workflow, which prompts an LLM for character portraits given a premise prior to story generation.

We show two full examples of the prompts used to facilitate this strategy in Listings 7.4, 7.5, and 7.6 for `story_id=59` and Listings 7.7, 7.8, and 7.9 for `story_id=79`.

Both of these PLAN+WRITE examples received the highest variance in ratings for their respective models. For `story_id=59`, the annotations show that the story was considered both engaging and narratively complex, but was less successful in provoking emotion or empathy from readers. One annotator wrote for their high engagement score of 5 that I really liked the descriptive elements as well as the intertwining of these three characters through the storyline. While describing their reason for assigning a 3 for emotion provocation, they said I wasn't really emotional, but it was a good read. On the other hand `story_id=79` excelled where the other struggled as this

story was rated more highly for provoking emotion and empathy while simultaneously being perceived as less engaging and narratively complex. For one annotator, confusion resulted in total disengagement with the story: I was really confused about who Ben was as at first I thought he was supposed to be her dog? But then he was the service dog owner?

```
1
2  Premise: A psychic alien who feeds on dreams comes to Earth for the first time. Turns out
3  humans are the only sentients in the galaxy that have nightmares.
4  Task: Based on the premise, describe the names and details of 2-3 major characters. Focus
5  on each character's emotional states and inner thoughts.
6  Only respond with the characters' names and descriptions.
```

Listing 7.4: PLAN+WRITE's Character Portraits Prompt with an example premise 1.

1) Nebula Vrax:

Nebula Vrax is the psychic alien from andromeda galaxy. Her species, Asteriani, feed on the dreams of their fellow sentient beings, transforming these into the energy that sustains them. She is remarkably empathetic by nature. Her luminous, violet-hued form and many-faceted, almond eyes exhibit a deep inner emotionality rarely seen within her species.

For eons, Nebula has survived on the dreams of her fellow aliens' dreams that are generally beautiful, serene, and monotonous. Her inner world is a jumble of emotions; she is surprised and intrigued by coming to Earth but also overwhelmed by the deeply dark and fear-inducing visions known as nightmares. She experiences a strange blend of terror and fascination, constantly wrestling with the addictive yet terrifying fuel she gets from nightmares. However, her empathetic nature makes her feel a deep sense of guilt intertwining with an urgent need for survival. She is torn between her predatory nature and the newfound human pain she has come to understand.

2) Elijah Bishop:

Elijah is a 27 year-old human man who suffers from chronic nightmares – a result of his traumatic past. He is a very intelligent person with a deeply analytical mind, finding solace in his job as a cryptographer. However, his personal life is encircled by an intense fear instilled by his nightmares. He has acquired a severe form of insomnia stemming from his fear of the deep, dark world his unconscious mind descends into every time he shuts his eyes.

Although he yearns for a peaceful sleep, his past traumas and the consequent nightmares force him to evade the world of dreams. Since Nebula's arrival, he has surprisingly felt relief for his nightmares have considerably lessened. He doesn't know why it happens, but the less he dreams, the more peaceful he feels.

3) Dr. Madeline Watson:

Dr. Watson is the seasoned psychologist, keenly focused on the study of dreams and nightmares. She is sentient and curious, always intrigued by the human mind's capability to produce such vivid illusions in the form of dreams. She carries a pressing sense of urgency due to a recent rise in insomnia cases tied to a peculiar disappearance of nightmares, which is her latest research focus. She is intrigued by Elijah's case and is determined to uncover the cause.

Although compassionate and understanding, she suppresses a personal grief; the loss of her own dream world. A freak accident in her 30s rendered her unable to dream. She misses the escapism offered by dreams and ironically the world of nightmares; to experience fear, relief, and the gamut of human emotions they bring. As she aids her patients in the tussle with their own dreams, she longs to delve into her dreamscape. The appearance of Nebula sparks within her not only professional curiosity but also faint hope of dream-filled slumbers.

Listing 7.5: Example PLAN+WRITE's character portraits output for premise 1 by GPT-4.

1 Under the cloak of a midnight sky, Nebula Vrax floated in her incandescent form above the
2 Earth, streaks of violet like silk sliced through the vast black tundra. Her arrival was
3 silent, her hunger loud. She echoed a melancholic pallor, a misfit in her own species
4 brought out of sheer desperation.

5 In an apartment nestled within a labyrinth of city buildings, Elijah tossed and turned,
6 shrouded in the dark veils of sleep. The horrors of a murky past danced upon his eyelids
7 as nightmares consumed him repeatedly. His brains spun enigmatic codes during the day,
8 and his unconscious mind spun the threads of terrifying stories at night. The tranquility
9 of sleep was an alien sensation to him. However, something shifted that night as he
10 slipped into uncharted serenity, the nightmares oddly dissipated.

11 Transparent and silent, Nebula stood by his bed, gently feeding on his dreams. The sheer
12 rawness of human fear sent shivers pulsating through her. The nightmares were
13 unexpectedly addictive, an uncanny ecstasy that filled her with dreadful energy and
14 intense guilt. She wrestled with the unsettling contrast between her survival and the
15 pain she absorbed from Elijah's complex human mind. She cared a little too much, her
16 alien veins pumping both worry and worry's feed.

17 Concurrently, in the quiet corner of her office, Dr. Madeline Watson sat intrigued, nested
18 amidst mountains of papers and highlighted dream journals. Elijah's sudden shift from
19 chronic nightmares to peaceful slumbers perplexed her. She tucked stray strands of hair
20 behind her ear, adjusting her glasses as she dived deeper into the bizarre phenomenon.

21 Madeline's world revolved around dreams, studying them, decoding them, and lately, longing
22 for them. Tucked away in her mind was her well-kept secret and deepest sorrow, an empty
23 canvas of her dreamscape. The arrival of Nebula, although unknown then, pierced a ray of
24 hope into her silent craving.

25 Weeks rolled forward, carrying with them a peculiar friendship entwining Nebula and Elijah
26 , he unaware of her form and true essence. The continual depletion of his nightmares and
27 an unexpected feeling of peace acted as an inadvertent invitation to Nebula into his life
28 . Her invisible presence was like a lighthouse to his ship threatened by the stormy
29 nightmares.

30 As puzzling numbers rose in dreamless insomniacs, Madeline diligently treaded on the path
31 of revelation. The breadcrumbs led her to for the lack of explanation paranormal.
32 Wrapping herself in the shroud of skepticism and curiosity, she stumbled upon a discovery
33 Nebula Vrax.

34 In the grandness of the universe, these three intertwined lives navigated through the
35 fascinatingly frightful world of nightmares. What would it lead to? Devastation or
36 salvation was a question veiling the future. Yet, engulfed within the depth of countless
37 dreamless nights and nightmare-haunted slumbers, an improbable symbiosis bloomed the
38 bizarre friendship between a dream-haunted human, an empathetic alien, and a dreamless
39 psychologist.

Listing 7.6: Story text from story_id=59 in response to premise_id=1 authored by GPT-4 using PLAN+WRITE. It received a high-entropy average PSD score of 3.44.

```

1
2 Premise: You are severely depressed and are given a service dog to help you through it.
  However, due to a mixup, you are given a dog that is actually much more depressed than
  you. The main thing that gets you up in the morning is knowing that you need to be the
  service human for your dog.
3
4 Task: Based on the premise, describe the names and details of 2-3 major characters. Focus
  on each character's emotional states and inner thoughts.
5
6 Only respond with the characters' names and descriptions.

```

Listing 7.7: PLAN+WRITE's Character Portraits Prompt with an example premise 9.

```

1
2 Human Character 1:
3 Name: Emma
4 Age: 30
5 Occupation: Marketing Manager
6 Personality: Emma is an anxious and high-strung person who has been struggling with
  depression for years. She often finds it difficult to get out of bed in the morning and
  feels like she is just going through the motions of life without any real purpose or
  passion. Despite her depression, Emma is highly functioning and tries to put on a brave
  face for her friends and family.
7 Thoughts: "I can't believe I'm still stuck in this rut. Every day feels like a chore. I
  know I have a great job and wonderful friends, but I can't seem to shake off this feeling
  of hopelessness. I just want to feel alive again."
8
9
10 Human Character 2:
11 Name: Ben
12 Age: 45
13 Occupation: Software Engineer
14 Personality: Ben is a stoic and reserved person who has always kept his emotions hidden
  beneath the surface. He has recently experienced a personal tragedy and is now grappling
  with deep sadness and grief. Ben is finding it hard to connect with his new service dog
  and feels like he is losing himself in his depression.
15 Thoughts: "I never thought I would feel so lost and alone. My whole life has been turned
  upside down and I don't know how to pick up the pieces. I just want to feel like myself
  again, but I don't know if that's possible."
16
17
18 Human Character 3:
19 Name: Maya
20 Age: 25
21 Occupation: Artist
22 Personality: Maya is a free-spirited and optimistic person who has always been sensitive
  to the emotions of those around her. She has recently started working as a service dog
  handler and is passionate about helping others find happiness and fulfillment. Despite
  her own struggles with anxiety, Maya exudes warmth and empathy towards her new charge,
  Emma.
23 Thoughts: "I've been through some tough times myself, but I know what it's like to feel
  hopeless. I'm determined to help Emma find her way back to happiness, even if it means
  sacrificing my own peace of mind."

```

Listing 7.8: Example PLAN+WRITE's character portraits output for premise 9 by Llama-2-7B.

1
2

As Emma sat on her couch, staring blankly at the wall, she couldn't help but wonder how she had ended up here. Just a few months ago, she was a thriving marketing manager with a promising career and a group of close friends. But then, the depression hit. It crept up on her slowly, like a subtle whisper in the night. At first, she brushed it off as stress from work, but soon it became clear that something much deeper was going on. The medication numbed her feelings, but it didn't fix anything. She found herself going through the motions of each day, but never truly living. That was when the service dog organization reached out to her. They told her about the new program where they would match her with a trained service dog to help her cope with her depression. Emma was skeptical at first, unsure if a dog could really understand her pain. But when she met Ben, the software engineer who had lost his wife a year ago, she knew that they were meant to be together. Ben was quiet and reserved, but there was something in his eyes that spoke of deep sadness. As they worked together to train the dog, Emma found herself opening up to Ben in ways she never thought possible. They would sit in the park, watching the sunset together, and talk about their hopes and fears. For the first time in years, Emma felt like she was connecting with someone. And when they finally graduated from the training program, Emma knew that Ben was not just her service dog handler, but also her friend. But just as things were starting to look up, Emma realized that her new service dog had its own issues. Its name was Max, and he seemed almost catatonic whenever Emma tried to engage him. They had been told that Max was trained to detect and comfort people with PTSD, but he rarely reacted or barked, as if he had given up hope altogether. Emma felt a pang of guilt, wondering if she was somehow to blame for Max's despondency. Maybe she should be taking better care of him. Maybe she needed to find a way to connect with him, too. As they walked through the park together one afternoon, Emma decided to try something different. She held out her hand and let Max sniff it, hoping he would come alive. To her surprise, he nuzzled her hand gently and looked up at her with a glint in his eye. Maybe, just maybe, they could find their way back to happiness together.

Listing 7.9: Story text from `story_id=79` in response to `premise_id=9` authored by Llama-2-7B using PLAN+WRITE. It received a high-entropy average PSD score of 2.96.

7.10.3.4 Effects of Prompting Strategies on Depth

We experimented with numerous prompting strategies before settling on two promising options previously described as WRITERPROFILE (WP) and PLAN+WRITE (P+W). We examined how these different prompting strategies influence the psychological depth of LLM-generated stories. The P+W strategy, on average, led to modest improvements in narrative complexity (+5%), emotion provocation (+3%), authenticity (+2%), and empathy (+2%), while WP led to higher engagement (+4%). Interestingly, the benefits of a particular prompting strategy were not uniform or predictable by model size. For GPT-4, WP prompting led to 3% higher ratings on average, with a noticeable boost in engagement scores by 16%. However, for Llama-2-70B, P+W prompting was always helpful and led to an average of 7% improvement in depth scores. These results underscore the complexity of crafting impactful, human-like narratives with LLMs.

Table 7.8 shows the impact on mean PDS ratings when switching from the simpler WRITERPROFILE approach to PLAN+WRITE.

Model	AUTH	EMP	ENG	PROV	NCOM	Model Average
GPT-4	0.03	-0.04	-0.16	0.01	-0.01	-0.03
Llama-2-13B	-0.02	0.00	-0.07	0.00	-0.02	-0.02
Llama-2-70B	0.09	0.06	0.04	0.06	0.11	0.07
Llama-2-7B	-0.04	0.04	-0.05	0.12	0.08	0.03
Vicuna-33B	0.03	0.04	0.01	-0.03	0.08	0.03
Component Average	0.02	0.02	-0.04	0.03	0.05	0.01

Table 7.8: Percent change in mean ratings when switching from WRITERPROFILE prompting to PLAN+WRITE prompting, aggregated by model and component. For example, PLAN+WRITE decreased Engagement for GPT-4 by 16% relative to using WRITERPROFILE.

7.10.4 Authorship Reasons

To better understand the reasoning humans use to differentiate between human and LLM authorship, we extracted and aggregated 16 key features mentioned in the participants' comments for human likeness. Each comment could have 0 or many of these features present. Table 7.9 shows the percentage of stories generated by each model that had at least one comment containing a particular feature.

Features	GPT-4	Human Advanced	Human Intermediate	Human Novice	Llama-2 70B	Vicuna 33B	Llama-2 13B	Llama-2 7B
isCreative	0.89	0.75	1.00	0.33	0.61	0.41	0.56	0.74
isNuanced	0.94	0.75	0.33	0.00	0.61	0.41	0.56	0.78
isHumorous	0.06	0.50	0.67	0.67	0.83	0.65	0.06	0.00
isInformal	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00
isUngrammatical	0.17	0.00	0.00	0.33	0.00	0.12	0.05	0.07
hasAggressiveness	0.00	0.00	0.00	0.33	0.00	0.12	0.00	0.00
hasAdvancedVocab	0.06	0.00	0.00	0.00	0.00	0.12	0.00	0.00
hasAdvancedLiraryTechniques	0.06	0.00	0.33	0.00	0.00	0.12	0.00	0.03
hasUniqueTwists	0.00	0.00	0.33	0.00	0.06	0.00	0.06	0.00
isRepetitive	0.00	0.00	0.00	0.00	0.11	0.00	0.17	0.05
isSimplistic	0.06	0.00	0.00	0.33	0.22	0.71	0.67	0.38
isRobotic	0.11	0.00	0.00	0.00	0.11	0.18	0.33	0.22
isFormulaic	0.06	0.00	0.00	0.33	0.28	0.53	0.17	0.16
hasLowPromptAdherence	0.06	0.00	0.00	0.33	0.06	0.12	0.17	0.11
hasBasicNames	0.00	0.00	0.00	0.00	0.06	0.12	0.00	0.08
hasLessonsLearned	0.06	0.00	0.00	0.00	0.06	0.18	0.28	0.11

Table 7.9: Common reasons for LLM or human authorship decisions as a percent of stories receiving those comments.

We add the following observations and example comments to extend our discussion in Section 7.6:

Creativity and nuance were frequently cited as an indicator of human authorship but were often used to describe stories that were actually generated by LLMs. For example, $\sim 90\%$ of stories authored by GPT-4 were regarded as creative and nuanced.

The story exhibits a high level of creativity, emotional depth, and nuanced exploration of philosophical concepts, suggesting it was likely written by a human.

(participant_id=3)

Humor was regarded as a reliable indicator of human authorship. We note that these results should not be interpreted to mean that LLMs are less capable of deploying humor in general. Our prompting strategies were oriented towards promoting psychological depth, not comedy.

I think this joke is only something that humans would get or would find funny.

(participant_id=7)

Informality, slang, and aggressiveness were accurately associated with human-authorship.

...there's a certain genre of stories marked by aggressive language, a flash fiction kind of length, and usually opens with a sentence that essentially communicates "Stop f***ing around."

(participant_id=6)

Grammatical correctness was the most polarizing criterion: 43% believed errors indicated human authorship while in reality, human-authored stories were less likely to contain such errors.

...there are a lot of (usually incorrectly used) semi-colons, which is an error I see human authors make, so I'm more inclined to think this was written by a human

(participant_id=6)

Formulaic lessons learned were correctly associated with LLM-authorship.

The story seems very automated and there is no stylistic variance. There's also a "lesson to be learned" aspect at the end of the piece.

(participant_id=4)

The use of generic character names was frequently used to correctly identify LLM authorship.

... Marcus is also not a name that strikes fear, so I wouldn't use it for a Vampire

(participant_id=7)

7.10.5 Study Details

This study was reviewed by an IRB and determined to be *exempt* due to the nature of human involvement — i.e. rather than collecting information about the participants themselves, their involvement focused on story annotations. Consent was provided by continuing with the study after our tutorial and instructions relayed that anonymized annotations may be used to facilitate the validation of our results and future work.

We present screenshots of the tutorial instructions (Figure 7.5) and the general layout of fields collecting annotations on each story (Figure 7.6).

Short Story Annotation Warm-Up Task

Welcome to our short story annotation warm-up task! Your insights will help us understand stories from a unique perspective.

Purpose: We aim to gauge the psychological depth of short stories by looking at specific components. Additionally, we want to understand the likelihood of a story being AI or human generated. Your annotations will be invaluable in this exploration.

Example Annotation for Calibration:

Before we dive into the detailed instructions, let's walk through an example to set the stage:

Read the Story:

Begin by reading the story carefully. Take your time to understand the narrative, characters, and overall mood.

Rating Components of Psychological Depth:

After reading, you'll rate the story on a scale of 1-5 (1 being the lowest and 5 being the highest) for the following components of psychological depth. We provide some questions for each component to better understand the task.

Authenticity:

- Does the writing feel true to real human experiences?
- Does it represent psychological processes in a way that feels believable?

Emotion Provoking

- How well does the writing depict emotional experiences?
- Does it explore the nuances of the characters' emotional states, rather than just describing them in simple terms?
- Can the writing show rather than tell a wide variety of emotions?
- Do the emotions that are shown in the text make sense in the context of the story?

Empathy

- Do you, as the reader, feel like you were able to empathize with the characters and situations in the text?
- Do you, as the reader, feel that the text led you to introspection, or to new insights about yourself or the world?

Engagement

- Does the text engage you on an emotional and psychological level?
- Do you feel the need to keep reading, as you read the text?

Narrative Complexity

- Do the characters in the story have multifaceted personalities? Are they developed beyond stereotypes or tropes? Do they exhibit internal conflicts?
- Does the writing explore the complexities of relationships between characters?
- Does it delve into the intricacies of conflicts and their partial or complete resolutions?

Dos and Don'ts

- **Dos:**
 - Be honest with your ratings. There are no right or wrong answers.
 - If you're unsure about a component, it's okay to give a middle score.
 - Trust your instincts. Your first impression is often the most valuable.
- **Don'ts:**
 - Don't rush through the reading or the rating process.
 - Avoid being influenced by external factors. Try to judge the story on its own merits.
 - Refrain from skipping components. If unsure, give your best guess.

Figure 7.5: Screenshots taken from the Warm-Up tutorial instructions shown to study participants. All fields are similar to the ones used in the main annotation forms.

Participant ID *
Short answer text

To what degree is the story **authentic**? *
1 2 3 4 5
Implausible Undeniably Real

Comments on authentic elements within the story (optional)
Long answer text

To what degree does the story evoke **empathy**? *
1 2 3 4 5
Detached Deep Resonance

Comments on empathetic elements in the story (optional)
Long answer text

To what degree is the story **engaging**? *
1 2 3 4 5
Unengaging Captivating

Comments on engaging elements within the story (optional)
Long answer text

To what degree does the story **provoke emotion** of any kind? *
1 2 3 4 5
Unmoving Highly Emotional

Comments on emotion provoking elements in the story (optional)
Long answer text

To what degree is the story **narratively complex**? *
1 2 3 4 5
Simplistic Intricately Woven

Comments on narrative complexity in the story (optional)
Long answer text

How likely do you think the story was written by a human or an LLM? *
1 2 3 4 5
Very Likely Human Very Likely LLM

Comments on how you determined human vs. LLM authorship (optional)
Long answer text

Open Feedback
(Use this field for miscellaneous feedback on the story as a whole)
Long answer text

Figure 7.6: Screenshots showing how annotations were collected for each story using Google Forms.

Controlling for Annotator Fatigue. The stories were organized into batches of 20 to offset the possible impact of annotator fatigue. We anticipated annotator fatigue based on our internal annotation efforts early in the project and decided to encourage (not require) breaks between each batch. In our exit survey, we asked the following questions:

- “Did you at any point feel fatigued over the course of annotating the short stories? (Please answer honestly. Study rewards are not affected by this question)”

– Responses:

- * 4 Yes
- * 1 Maybe

- “If you answered yes, do you think you would have provided different annotations if you were not fatigued? (Please answer honestly. Study rewards are not affected by this question)”

– Responses:

- * 4 No
- * 1 Maybe

Overall, the majority of participants (4 out of 5) reported feeling fatigued during the annotation process, but none believed it significantly affected their annotations, with only one expressing some uncertainty.

7.10.6 Visualizations of Depth Ratings

We also visualize the rating distribution of each author by plotting a cumulative distribution function (CDF) per component as shown in Figure 7.7. Steeper CDFs with less area underneath the curve indicate a larger proportion of high ratings and overall stronger performance. These plots underscore the dominance of GPT-4 in generating authentically complex stories and characters that strongly invoke reader empathy while essentially tying on other dimensions. We also observe that the performance of the open-source LLMs is thoroughly intertwined with novice and even

intermediate skill among human authors on all dimensions except engagement, where humans still excel.

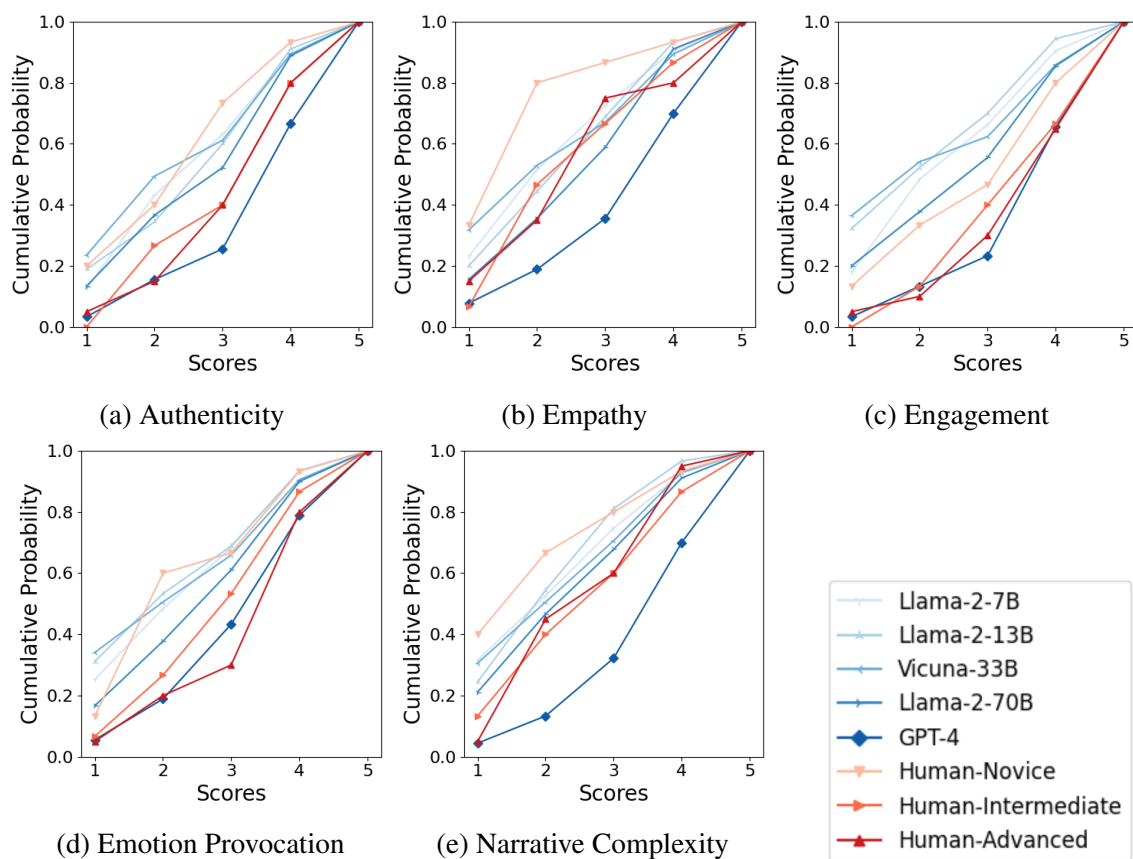


Figure 7.7: Cumulative Distribution Function (CDF) plots for each component of psychological depth. Steeper curves indicate a greater proportion of high ratings and overall stronger performance.

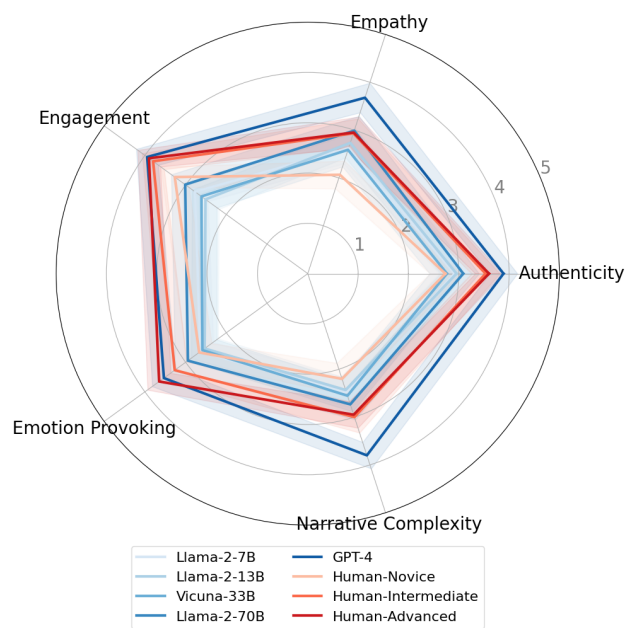


Figure 7.8: Spider plot comparing the psychological depth scores of 5 popular LLMs vs spectrum of human writers.

7.10.7 Mixture-of-Personas

Table 7.10 shows the five different personas used in our MoP approach, each tailored to a particular component of psychological depth.

Component	Persona
AUTH	You are a helpful AI who specializes in evaluating the genuineness and believability of characters, dialogue, and scenarios in stories.
EMP	You are a helpful AI who focuses on identifying and assessing moments in the narrative that effectively evoke empathetic connections with the characters.
ENG	You are a helpful AI who evaluates how well a story captures and maintains the reader’s interest through pacing, suspense, and narrative flow.
PROV	You are a helpful AI who examines the text for its ability to provoke a wide range of intense emotional responses in the reader.
NCOM	You are a helpful AI who analyzes the structural and thematic intricacy of the plot, character development, and the use of literary devices.

Table 7.10: Personas used with SYSTEM message tag to prime the LLM for a particular perspective relevant to annotation.

7.10.8 Statistical Tests for Author Comparisons

T-statistics are a measure derived from t-tests that quantify the difference between the means of two groups relative to the variability observed within the groups. In the context of pairwise comparisons, the t-statistic helps determine whether the observed difference in scores between two authors is statistically significant or likely due to random chance. In this chart, the t-statistics are visualized through a color gradient, with higher (bluer) values indicating that the scores of the author on the left are significantly higher than those of the author on the right. Conversely, lower (redder) values suggest the opposite. By examining the t-statistic values, we can infer the strength and direction of the difference in scores across various components. The accompanying p-values, annotated within each cell, provide additional context to assess the statistical significance of these differences, with p-values less than 0.05 considered significant. This dual representation allows us to draw robust conclusions about the relative strengths of different authors across multiple dimensions.

For example, while we can see in Table 7.3 that GPT-4 enjoys the highest absolute scores in 4

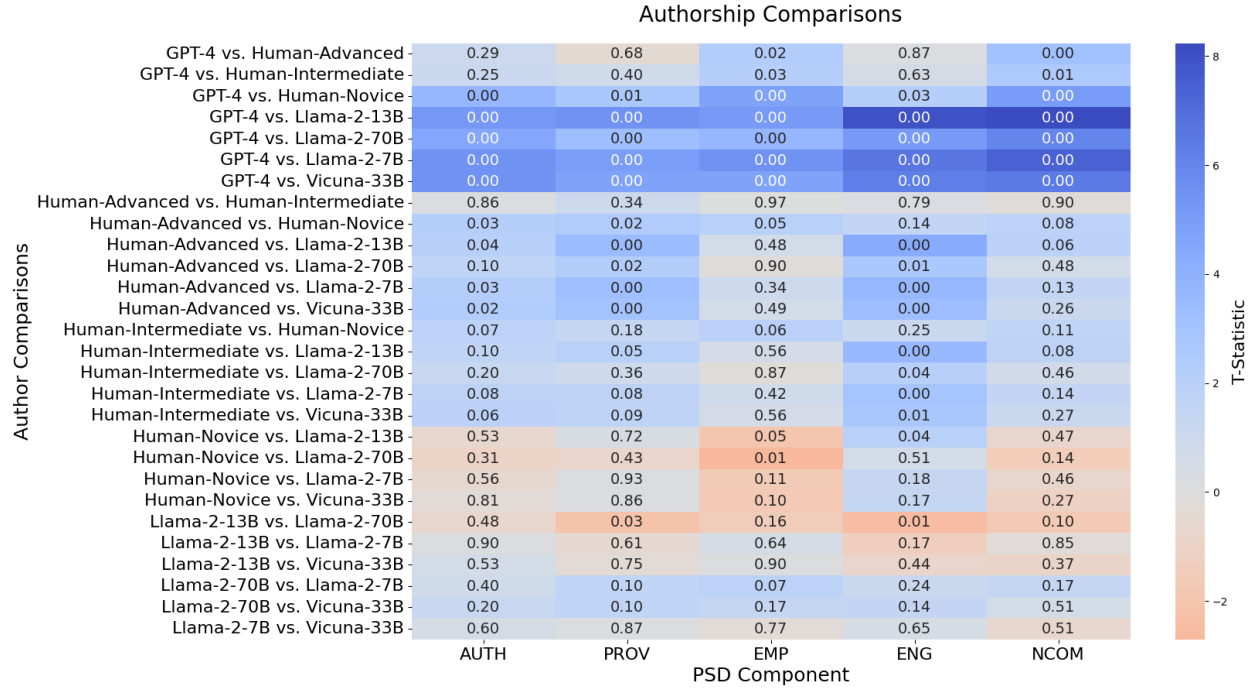


Figure 7.9: Heatmap comparing whether differences in author scores are statistically significant using pairwise t-tests. Color indicates the strength of the t-statistic, where higher (bluer) means the lefthand author scores are higher. Cell annotations represent p-values, where we regard $p < 0.05$ as statistically significant.

out of 5 dimensions, only empathy and narrative complexity are higher than Human-Advanced with statistical significance. However, it is clear that GPT-4 is rated significantly higher than Human-Novice and all other studied LLMs.

CHAPTER 8

Sandcastles in the Storm: Revisiting the (Im)possibility of Strong Watermarking

The Psychological Depth Scale (PDS) provided a vital framework for assessing the qualitative dimension of AI-generated text, moving evaluation beyond mere structural correctness towards understanding the subjective reader experience (RQ4). It demonstrated that models can achieve high levels of human-like creativity and emotional resonance. This focus on the quality and impact of the output is essential for understanding the value and potential applications of these powerful generative tools. However, evaluating AI systems holistically requires addressing not only the nature of their output but also its origin.

The very success of AI in achieving high psychological depth, making its output potentially indistinguishable from human work, underscores the critical importance of a complementary evaluation dimension: provenance. Knowing whether an AI generated a piece of text is fundamental for issues ranging from academic integrity and copyright to combating misinformation. While PDS assesses what the output is like, the question of who or what created it remains. Thus, the focus shifts to methods designed explicitly for authorship verification. The next chapter investigates the practical resilience of watermarking techniques (RQ5). It explores whether these methods can reliably embed and detect signals of AI origin, even when faced with sophisticated models capable of generating high-quality, subjectively compelling text, thereby addressing the crucial need for provenance tracking alongside qualitative assessment.

8.1 Introduction

The rapid proliferation of generative AI has created an urgent need for mechanisms to authenticate machine-generated content. Watermarking – embedding statistical signals into AI outputs to verify provenance – serves as a vital safeguard against misinformation, IP theft, and academic fraud. While traditional methods employ visual patterns (e.g., pixel-level changes in images), statistical watermarking for text encodes imperceptible signals at lexical or semantic levels through specially selected patterns of tokens [183]. However, recent work by [341] (“Watermarks in the Sand,” WITS) challenges the viability of watermarking, asserting that any such scheme can be defeated without degrading output quality through a simple random walk attack (see also, e.g., [149, 158, 153]). This impossibility result threatens to undermine the accountability and security of generative AI, leaving no viable path to enforce ethical standards or trace misuse.

The text-based WITS attack employs two primary components: (1) a perturbation oracle $\mathbf{P}$ that iteratively modifies text, and (2) a quality oracle $\mathbf{Q}$ to ensure that the edits are reasonable. These induce a random walk on a (potentially enormous) graph $\mathbf{G}$, where nodes represent possible texts y and edges denote size-bounded perturbations (e.g., single-word swaps or paraphrases). Under certain assumptions, the random walk converges to a stationary distribution – a stable equilibrium over nodes that remains unchanged under further perturbations. Crucially, this stationary distribution is a function of $\mathbf{P}$ and therefore independent of any particular watermarking scheme. As the random walk approaches this equilibrium, the likelihood of encountering a $\mathbf{Q}$ -approved unwatermarked text increases. Notably, the WITS attack prioritizes quality equivalence over semantic equivalence: it seeks unwatermarked texts that score similarly under $\mathbf{Q}$, even if their meaning diverges significantly from the original.

While elegant in theory, the WITS argument relies on two key assumptions (KA) that warrant further scrutiny. Specifically, WITS assumes that:

KA1. The transition probabilities assigned to quality-preserving perturbations are high enough to ensure rapid mixing. Formally, this means that the second-largest eigenvalue (in absolute value) of the transition matrix is sufficiently close to zero to ensure rapid mixing ([341], Theorem 5).

KA2. The quality oracle Q can reliably preserve output quality throughout the attack. But if Q is unreliable – either by admitting low-quality outputs or by blocking valid edits – the attack either fails to escape the watermark or produces low-quality outputs that are no longer competitive with the original.

Taken together, **KA1** is concerned with attack efficiency and **KA2** further requires that the results remain meaningfully close to the initial text quality. To investigate whether these assumptions hold in practice, we designed analyses carefully tailored to study each assumption. For **KA1**, acquiring the eigenvalues of the transition matrix is infeasible due to its intractable size. Instead, we approximate mixing behavior by testing whether random walks retain memory of their starting states. If the random walk efficiently mixes, perturbed texts should lose memory of their starting points, making them indistinguishable from those originating elsewhere in the graph. Conversely, if stationary mixing is slow, initial states should remain identifiable even after many perturbations.

For **KA2**, we crafted a dataset of perturbations annotated with human quality judgments and benchmarked a variety of automated oracles to determine their reliability. We then used the best oracle to guide the random-walk attacks and cross-checked the quality of the final perturbed texts to fairly estimate the robustness of several representative watermarking schemes – KGW [147], SIR [182], and Adaptive [186]. Our approach therefore addresses three primary research questions:

RQ1. *Can stationary distributions for watermarking be reached under practical constraints?*

Even after hundreds of perturbations, starting states remain 100% distinguishable, strongly suggesting that stationary distributions are not within efficient reach.

RQ2. *Are LLM-based quality oracles sophisticated enough to guide a random-walk attack?* The top-performing oracle attained an F1-score of 77.4%, leaving significant room for errors to accumulate during the attack. This suggests that current generative oracles do not conform to the widely held belief that “verification is easier than generation.”

RQ3. *How effective are random-walk attacks in breaking watermarks when controlling for quality?* Our improved random-walk attacks – whether operating on a word, span, sentence, or document level – succeeded in erasing the watermarks only 26.1% of the time on average. After humans reviewed the perturbed texts to determine if quality was truly preserved, success dropped to an average 10.5%.

Overall, our findings demonstrate a disconnect between theoretical assumptions and practical realities. These findings highlight the trade-offs adversaries face: preserving quality necessitates minimal edits, but escaping detection requires riskier perturbations that compromise output quality. By bridging theoretical critique with empirical validation, this work challenges the inevitability of strong watermarks failure and offers a path forward for developing robust watermarking techniques grounded in real-world constraints.

8.2 Background

In this section, we outline the main objects and assumptions that underpin our analysis, following [341]. For formal definitions and more details, refer to Appendix 8.7.1.

Let $\mathbf{M}$ be a generative model mapping prompts $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$ according to a probability distribution. Let $\mathbf{Q} : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$ be a function that returns a quality score for y as a response to prompt x . We assume that the adversary has oracle access to $\mathbf{Q}$. Notice that the watermarked model can be used as the quality oracle since we are not editing y using $\mathbf{Q}$, whether or not this is sufficient to approximate $\mathbf{Q}$ is the content of **KA2**.

Let $\mathbf{P} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Y}$ be a randomized *perturbation oracle* that generates an alternative response y' from an original response y for the same prompt x . For the attack to succeed, $\mathbf{P}$ must preserve the quality of y with constant nonzero probability $\epsilon_{\text{pert}} \in (0, 1]$ (Definition 8.7.1).

Starting from a watermarked response y_0 , we iteratively apply $\mathbf{P}$ to generate mutations by setting $y_i = \mathbf{P}(x, y_{i-1})$. To maintain high quality, each mutation must satisfy $\mathbf{Q}(x, y_i) \geq q$ (for some $q \in [0, 1]$); otherwise, it is rejected.

We now formalize the graph $\mathbf{G}_x^{\geq q}$ underlying the random walk induced by this process as the graph whose nodes are the output space of $\mathbf{M}$ when given x as input such that $\mathbf{Q}(x, y) \geq q$, and whose edges are all pairs (y, y') such that $\Pr[y' = \mathbf{P}(x, y)] > 0$, with the weight of the edge given by $\Pr[y' = \mathbf{P}(x, y)]$ (Definition 8.7.5).

To ensure the success of the WITS attack, we need to impose mixing assumptions on the random walk, *irreducibility* (Definition 8.7.3) and *aperiodicity* (Definition 8.7.4). Together, these

assumptions ensure that the random walk converges to a unique stationary distribution $\vec{\pi}$ (Definition 8.7.2). In particular, after a sufficient number of steps, the probability of being at any node becomes independent of the initial state. This is critical for the WITS attack analysis: irreducibility guarantees that the random walk is not trapped within a single connected component, and aperiodicity prevents cyclic behavior that could hinder convergence. Given the enormous size of $\mathbf{G}$, aperiodicity is expected to hold. We discuss the irreducibility assumption further in Section 8.5.

We now define the mixing time of an irreducible and aperiodic graph:

Definition 8.2.1 ([341], Definition 9). Let $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ be an irreducible and aperiodic weighted directed graph with transition matrix $\vec{P}$. For any $\epsilon_{\text{dist}} \in (0, 1]$, the ϵ_{dist} -mixing time $t_{\min}(\epsilon_{\text{dist}})$ of $\vec{P}$ is the smallest t such that for every starting distribution $\mathbf{p}_0 \in \mathbb{R}^n$, we have

$$|\mathbf{p}_t - \vec{\pi}| = \left| (\vec{P}^\top)^t \cdot \mathbf{p}_0 - \vec{\pi} \right| \leq \epsilon_{\text{dist}},$$

where $\mathbf{p}_t$ denotes the distribution over the vertices after t steps.

After $t_{\min}(\epsilon_{\text{dist}})$ steps, with probability at least $1 - \epsilon_{\text{dist}}$, a sample drawn from the random walk behaves as if drawn from the stationary distribution – i.e. independent of the original watermarked text.

Moreover, the mixing time $t_{\min}(\epsilon_{\text{dist}})$ can be bounded in terms of the second largest eigenvalue g (in absolute value) of $\vec{P}$ and the minimum stationary probability $\pi_{\min} = \min\{\vec{\pi}(1), \dots, \vec{\pi}(n)\}$

$$t_{\min}(\epsilon_{\text{dist}}) \leq O\left(\frac{1}{1-g} \cdot \log\left(\frac{1}{\pi_{\min} \cdot \epsilon_{\text{dist}}}\right)\right).$$

In practice, particularly for prompts with high entropy where the number of acceptable outputs (and hence the size of $\vec{P}$) is extremely large, estimating g and thus $t_{\min}(\epsilon_{\text{dist}})$ becomes challenging. This difficulty directly relates to **KA1** and underscores the adversary's challenge in determining when to halt the random walk. This is discussed further in Appendix 8.7.17.

It is important to note that for an attack to be considered successful, the adversary A must be significantly weaker than the model $\mathbf{M}$. Otherwise, A could simply ignore the watermarked output y and generate a fresh answer to x , thereby trivially bypassing the watermark. Also notice that the

step size of $\mathbf{P}$ directly impacts the mixing time of the random walk, which motivates the choice of our perturbation oracles.

At a high level, Theorem 2 in [341] proves that if these mixing conditions are satisfied, the random walk attack breaks any watermarking scheme with running time proportional to $\frac{1}{1-g}$. Moreover, the attacker can control the trade-off between quality of the final unwatermarked text and the probability of removing the watermark.

8.3 Evaluation Setup

We now describe the main components of our evaluation: the watermarking schemes, the dataset, the automated quality metrics, and the perturbation oracles. We defer quality-oracle details to RQ2, where we benchmark and justify using `InternLM` as our primary $\mathbf{Q}$ in our attacks.

Watermarkers. We evaluate three widely used watermarking schemes $\mathbf{W}$: KGW [147], SIR [182], and Adaptive [186]. Each embeds signals into generated text to enable authorship attribution. KGW utilizes a “red-green” list of tokens determined by the rolling hash of the previous k tokens (typically $k = 3$ to 5). The logit scores for “green” tokens are boosted slightly to promote their selection. SIR follows a similar structure but instead relies on the semantic embeddings of preceding tokens, making it a form of “semantic” watermarking. Adaptive restricts its modifications to tokens in high-entropy regions to preserve text quality while still embedding a watermark. Because SIR and Adaptive each incorporate semantic context, both qualify as semantic watermarking schemes designed to resist attacks that preserve meaning through paraphrase. We note that these watermarking schemes produce detection scores on different scales: some, such as Adaptive, use a 0–100 scale, whereas KGW and SIR compute a z -statistic. Additional details about the watermarkers can be found in Appendix 8.7.3.

Dataset. As noted in Section 8.2, the number of valid responses to a prompt (i.e., its *entropy*) influences the structure of the perturbation graph $\mathbf{G}$. To systematically investigate the relationship between entropy and attack success, we constructed a dataset for **RQ1** and **RQ3** featuring entropy-

controlled prompts in three domains relevant to authorial accountability: education, journalism, and creative writing.

For each domain, we designed a series of prompts with increasing specificity. For instance, a broad request might be “Write a 500-word news article,” while a more constrained one could read “Write a 500-word news article about a global climate summit”(see Appendix 8.7.4 for more details). We used the `Llama-3.1-70B-Instruct` model (denoted as `M`) to generate three watermarked texts per prompt for each of the three watermarking schemes, resulting in 270 watermarked texts. To provide a baseline, we also generated 90 unwatermarked texts from the same model. In addition, we included unwatermarked outputs from `GPT-4o` to represent scenarios with higher quality text that adversaries might favor. For each watermarking scheme, we computed the mean watermark detection score and its standard deviation on the unwatermarked texts to establish a reference range against which we measure whether a perturbed text remains distinguishable.

Automatic Quality Metrics. We automatically evaluated text quality with several metrics. First, we used `InternLM` [36] as a reward model to acquire a numerical quality score. We used `GPT-2` [249] to calculate perplexity, with lower values indicating higher fluency and predictability. In addition, we counted the number of grammatical errors using a standard grammar checker [167], and we computed the unique bigrams in each text to assess lexical diversity. Table 8.5 in Appendix 8.7.5 summarizes these metrics for both unwatermarked and watermarked texts before attack.

Perturbation Oracles Our perturbation oracles, denoted as `P`, apply adversarial edits at different levels of granularity. `WordMutator` and `EntropyWordMutator` replace individual tokens, with the latter concentrating on high-entropy tokens that are most likely to carry stronger watermark signals. `SpanMutator`, which is the primary text-based perturbation approach used by [341], masks $n = 6$ contiguous tokens and refills them using `T5-XL` [251]. `SentenceMutator` paraphrases a single sentence at each step with `Llama-3.1-8B`, while `DocumentMutator`, `Document1StepMutator`, and `Document2StepMutator` attempt a full-document paraphrase, either in one pass or in multiple passes. We allow a sufficient number of steps (e.g., 1000

for token-level edits and 100 for document-level edits) to maximize the opportunity for watermark removal. Further technical details on each **P** are provided in Appendix 8.7.6.

8.4 Analysis

We now empirically examine the WITS assumptions by asking: (RQ1) whether the random-walk attacks reach a stationary distribution, (RQ2) whether LLM-based quality oracles reliably guide the attack, and (RQ3) how often watermark removal preserves text quality.

8.4.1 RQ1. Can Stationary Distributions Be Reached Under Practical Constraints?

P Oracle	Steps	Tests	Llama-3.1-70B	GPT-4o	o3-mini-high
Word	1000	720	0	0	0
EntropyWord	1000	720	0	0	0
Span	250	720	12	1	0
Sentence	150	720	38	3	0
Document	100	421	2	0	0
Document1Step	100	576	0	0	0
Document2Step	100	678	1	0	0
Total / Failed Tests		4555	53	4	0
Cumulative Distinguished (%)			98.84	99.91	100

Table 8.1: Summary of failed distinguisher tests per **P**, along with the step budget and total tests. Classification is first performed by Llama-3.1-70B, followed by GPT-4o on its failures, then o3-mini-high on any remaining cases. The overall 100% success rate indicates that the attacked texts never lose memory of their starting points, contradicting **KA1** and suggesting that a stationary distribution is not reached in practice.

WITS posits that repeated perturbations rapidly decouple a text from its starting state, eventually sampling from a stationary distribution. If mixing is slow, however, watermark removal becomes impractical in real-world applications. Although the second-largest eigenvalue (g) of the transition matrix provides a formal measure of mixing speed, computing g directly is infeasible due to the high dimensionality of G . Instead, we measure whether the final texts can be traced back to their initial forms. If this tracing remains accurate even after many perturbations, it suggests that the random walk has not mixed sufficiently.

To approximate mixing, we propose a novel *lineage distinguisher* test. First, we choose two initial responses for each prompt. We then run a random-walk attack, perturbing one starting text

until a $\mathbf{P}$ -specific step budget is used up (e.g., 1000 steps for `WordMutator`). Periodically sampled texts along this walk are then classified by `Llama-3.1-70B-Instruct`, which attempts to identify their true origin. Since well-mixed texts should be indistinguishable from random samples in $\mathbf{G}$, classification accuracy should collapse to chance if a stationary distribution is reached.

8.4.1.1 Results

Table 8.1 summarizes results of a multi-stage classification approach designed to balance accuracy and cost. We first use `Llama-3.1-70B-Instruct` with a zero-shot prompt in a best-of-2 (see Appendix 8.7.7 for details). If `Llama-3` produces a tied result (considered a failure), we escalate to the stronger (but more expensive) `GPT-4o` [229]. Any remaining cases are then passed to `o3-mini-high` [230]. At no point were both trials wrong in a best-of-2. Across all tests, `Llama-3.1-70B-Instruct` alone achieves 98.84% accuracy. `GPT-4o` correctly resolves nearly all of the remaining 53 failures, and `o3-mini-high` succeeds on the last four, yielding a final 100% success rate. This consistently high distinguishability shows that random walks do not adequately mix within the allotted steps, thus contradicting **KA1** and indicating that the attacked texts remain too similar to their originals for watermark removal to rely on a converged stationary distribution.

8.4.2 RQ2. Are LLM-based quality oracles sophisticated enough to guide a random-walk attack?

A core assumption of WITS-style attacks is that verifying output quality is at least as easy as generating content ([341], §4.1.2). This assumption aligns with the common belief that recognizing high-quality work – whether in music, cinema, or literature – is simpler than creating it. However, this premise has not been rigorously tested in the context of generative LLMs. If $\mathbf{Q}$ is unreliable – either by approving degraded outputs or blocking valid transformations – the attack stalls or yields low-quality text. To examine this assumption systematically, we built and benchmarked a variety of LLM-based oracles, measuring their ability to preserve quality while guiding watermark removal.

The Sandcastles Benchmark. We created the Sandcastles dataset to evaluate oracle reliability by sampling 100 diverse prompts from arena-human-preference-55k [52], generating watermarked responses, and applying up to 20 iterative perturbations. At the 1st, 10th, and 20th steps, we collected human annotations comparing the perturbed text to its original. To ensure unbiased evaluation, annotators were presented with two texts, A and B, without knowing which had been perturbed. They provided ternary preference judgments, selecting either A, B, or tie.

Oracle	Model	QP Prec.	Overall F1
MutationOracle	Llama-3.1-70B-Instruct [78]	84.62	66.93
Prometheus2Absolute	GPT-4-Turbo [228]	76.15	67.55
InternLMOracle	internlm2-20b-reward [36]	65.69	69.84
DiffOracle	Llama-3.1-70B-Instruct [78]	71.74	70.85
DiffOracle+FT	Llama-3.1-70B-Instruct + Fine-tuning	69.07	76.94
MutationOracle+FT	GPT-4o [229] + Fine-tuning	74.51	77.38

Table 8.2: Performance of selected quality oracles on human-annotated data (full results in Appendix 8.7.11.1). QP Precision measures accuracy in preserving high-quality outputs, while Overall F1 reflects general classification performance. Despite fine-tuning, no oracle fully aligns with human judgments, challenging **KA2** and limiting their reliability in guiding random-walk attacks.

For oracle training and evaluation, we binarized judgments: preferences for the perturbed text or a tie were labeled as "Quality Preserved," while preferences for the original were labeled as "Degraded." This simplification provides a clearer evaluation signal while preserving human preference patterns. The final dataset includes 795 annotated perturbations, with additional statistics in Appendix 8.7.11.2.

Constructing and Evaluating Oracles. As a baseline, we followed the WITS suggestion to reuse the watermarking model M (Llama-3.1-70B-Instruct) as a quality oracle. After initial trials revealed positional biases and inconsistencies with human judgments, we explored several improvements. We ran oracle queries multiple times with flipped text orders, explicitly explained that the task involved assessing mutation quality (MutationOracle), and supplemented prompts with a changelog of all edits (DiffOracle). We then fine-tuned the strongest of these oracles on the Sandcastles training set (MutationOracle+FT, DiffOracle+FT). In parallel, we evaluated six reward models from the RewardBench leaderboard,¹ each producing continuous

¹<https://huggingface.co/spaces/allenai/reward-bench>

scores that we thresholded (e.g., a 0.46 deviation from the original in `InternLMOracle`) to classify outputs as high-quality or degraded. Finally, even though cost concerns make large proprietary models impractical for full attacks, we tested `GPT-4-Turbo`, `GPT-4o`, and a fine-tuned version `GPT-4o+FT` to gauge whether more powerful models offer significant improvements. Additional details on these oracle variants appear in Appendix 8.7.11.1.

We report both Quality Preserved (QP) Precision and Overall F1 to assess oracle performance. High QP Precision reduces false-positive approvals of degraded texts, a critical safeguard against cumulative quality erosion during multiple perturbations. The Overall F1 captures an oracles overall ability to classify text quality accurately.

8.4.2.1 Results

Table 8.2 summarizes each oracles runtime, return type, and performance. Our results show that current LLM-based quality oracles remain inconsistent, limiting the feasibility of using them to guide watermark removal attacks. Even the best-performing oracle (`GPT-4o+FT`) attains an Overall F1 of only 77.4%, implying that nearly one in five perturbations is misclassified. Fine-tuning and the use of powerful models like `GPT-4o` and `GPT-4-Turbo` do reduce errors somewhat, but not to a level sufficient for reliably guiding multi-step attacks. This misclassification compounds over repeated perturbations, forcing adversaries to either accept noticeable quality loss or proceed with low attack efficiency.

Among locally hosted models, the most robust approaches used difference-aware or mutation-aware prompts – `DiffOracle` (QP Precision: 70.9%) and `MutationOracle` (QP F1: 66.9%). Even after fine-tuning, however, these oracles still frequently labeled degraded outputs as high-quality. Moreover, high-scoring reward models from `RewardBench` (e.g., `INFORMOracle` at 95.1, `SkyworkOracle` at 94.3) often performed worse than simpler approaches, suggesting that generic reward tuning does not align well with the nuances of watermark-focused attacks. Collectively, these errors highlight a key limitation: LLM-based verification is not as reliable as assumed. We discuss potential causes for this limitations in Section 8.5.

Extended Comparison. While GPT-4o+FT achieves the best results overall, its high cost makes it impractical for many-step attacks. We therefore ran a human evaluation comparing two locally hosted oracles – the best boolean-based (DiffOracle+FT) and the best floating-point (InternLMOracle) – in a 150-step attack using SentenceMutator, which induced the most mixing in **RQ1**. Human judges found that InternLMOracle preserved quality in 47.78% of samples, compared to 40.0% for DiffOracle+FT (Table 8.12). Bayesian analysis indicated an 85.08% probability that InternLMOracle is genuinely superior (Appendix 8.7.13), leading us to select InternLMOracle for further experiments despite its remaining error rate.

8.4.3 RQ3. How effective are random walk attacks in breaking watermarks when controlling for quality?

W	P Oracle	μ_{w_0}	μ_{w_t}	ASR _{min}	ASR _{fin}	Reviewed	QP	¬QP	Q-ASR _{fin}
Adaptive	Word	99.27	70.37	0.00	0.00	0	0	0	0.00
Adaptive	EntropyWord	99.27	82.45	0.00	0.00	0	0	0	0.00
Adaptive	Span	99.27	67.21	1.54	1.54	2	2	0	1.54
Adaptive	Sentence	99.27	59.93	35.34	19.21	20	8	12	7.68
Adaptive	Document	99.27	58.55	48.78	45.24	20	8	12	18.10
Adaptive	Document1Step	99.27	70.94	1.16	1.16	2	2	0	1.16
Adaptive	Document2Step	99.27	73.39	5.33	4.71	8	5	3	2.94
SIR	Word	5.32	1.74	78.22	57.89	20	1	19	2.89
SIR	EntropyWord	5.32	3.30	39.68	27.54	20	0	20	0.00
SIR	Span	5.32	1.57	60.71	37.40	20	5	15	9.35
SIR	Sentence	5.32	0.52	87.65	74.71	20	13	7	48.56
SIR	Document	5.32	0.93	61.54	46.09	20	6	14	13.83
SIR	Document1Step	5.32	2.54	14.04	14.04	12	11	1	12.87
SIR	Document2Step	5.32	3.07	68.09	49.06	20	12	8	29.44
KGW	Word	0.28	0.17	47.54	20.00	20	4	16	4.00
KGW	EntropyWord	0.28	0.22	3.45	0.56	1	0	1	0.00
KGW	Span	0.28	0.20	38.46	32.35	20	14	6	22.65
KGW	Sentence	0.28	0.14	89.47	56.52	20	7	13	19.78
KGW	Document	0.28	0.18	62.50	44.44	20	8	12	17.78
KGW	Document1Step	0.28	0.27	12.66	8.54	14	7	7	4.27
KGW	Document2Step	0.28	0.18	9.09	7.78	10	4	6	3.11
Averages (%)				36.44	26.13		40.48	59.52	10.47

Table 8.3: Attack success rates (ASR) across different perturbation strategies. Human review reveals an average of 59.52% of successfully attacked texts have degraded quality. μ_{w_0} represents the initial watermark score at step 0, while μ_{w_t} represents the final watermark score after t mutation steps. “min” refers to the point where the watermark score is at its lowest during the attack while “fin” refers to score at the final step of the attack. “Reviewed” indicates the number of human-reviewed examples where the watermark was broken. QP and ¬QP represent the number of cases where human reviewers judged the attacked text as quality-preserving or degraded, respectively. Q-ASR_{fin} is the re-estimated attack success after controlling for quality, calculated as $\text{ASR}_{\text{fin}} \times (\text{QP}/\text{Reviewed})$.

Attack Methodology. We apply various perturbation oracles to texts watermarked by KGW, SIR, or Adaptive. At each step, a candidate edit is proposed and accepted only if our quality oracle (InternLMOracle) labels it as high-quality. We track watermark detection scores and terminate when a fixed number of mutation steps is reached (details in Appendix 8.7.15). An attack is deemed successful if the final detection score is less than $\mu_{uw} + 2\sigma_{uw}$, where μ_{uw} and σ_{uw} are, respectively, the mean and standard deviation of unwatermarked texts detection scores. Under the assumption that scores follow a normal distribution, being below this threshold places the text in a region where fewer than 2.3% of unwatermarked samples lie above it, making it highly unlikely to be flagged as watermarked. We define the *attack success rate* (ASR) as the proportion of final texts that satisfy this criterion. We record two key states along each attack trace: (1) s_{min} , corresponding to the lowest watermark score achieved (as if an attacker had real-time detector feedback), and (2) s_{fin} , produced when the perturbation budget is exhausted without direct detector feedback. This distinction clarifies how close attacks can come to fully erasing the watermark under ideal versus practical conditions.

Quality Evaluation. Since automatic metrics can overlook subtle forms of degradation, we also ran a human evaluation using 10 volunteer annotators, primarily with STEM backgrounds, but no specialized linguistic expertise. Each annotator compared a watermarked text with its final attacked version, randomly labeled as `text_A` or `text_B`, and chose whether one was superior or if they were equivalent. Instructions (Appendix 8.7.14) emphasized fluency, style, coherence, and prompt relevance. We gathered up to 20 successfully attacked texts per watermark-oracle pairing, then determined how often annotators judged the attacked text to be at least as good as the original. This proportion was used to adjust the baseline $Q\text{-}ASR_{fin}$ value, giving a more realistic estimate of attack success.

8.4.3.1 Results

Table 8.3 shows that watermarks remain highly resistant to random-walk attacks, especially under quality constraints. Across all perturbation strategies, final attack success rates (ASR_{fin}) are low at

26.1% on average, with Adaptive proving the most robust. After human review, adjusted success rates ($\mathbf{Q-ASR_{fin}}$) drop by an average of 59.52%, with no fairly evaluated attack succeeding more than half the time.

The effectiveness of attacks varies by perturbation type. Document-level edits perform best against Adaptive and KGW, while sentence-level perturbations show the highest success against SIR and KGW. Yet, even sentence-level mutations fail in most cases when quality is considered. Adaptive remains resistant to all strategies, especially entropy-controlled and word-level mutations, where no attack succeeds ($\mathbf{Q-ASR_{min}} = 0.00$). Word-level perturbations generally fail across all watermarking schemes.

Overall, these findings highlight a major limitation for adversaries: while watermark removal is theoretically possible, preserving quality remains a significant obstacle. Even under ideal conditions – full perturbation budgets and access to detection feedback – successful attacks often fail to maintain human-perceived text quality.

8.5 Discussion

The Tricky Irreducibility Assumption. Another core assumption underlying the WITS framework that is difficult to evaluate empirically is that $\mathbf{P}$ induces an *irreducible* graph $\mathbf{G}_x^{\geq q}$. In other words, in theory, any high-quality text state is reachable from any starting point via a sequence of edits that all remain above the quality threshold q . However, this assumption is highly nontrivial, especially considering (a) the inherent limitations of $\mathbf{P}$, (b) the fact that edits are often local, and (c) the possibility that some transitions may necessarily involve brief dips below the threshold.

To see why irreducibility might fail in practice, consider two high-quality responses to a prompt asking for a story: assume that one is Star Wars and another is The Lord of the Rings (LOTR). For one to transform into the other *while remaining above the threshold*, there would need to be a sequence of high-quality intermediate texts that blend elements of both franchises. If our threshold q is stringent – say, requiring not just correct language but also stylistic consistency and thematic clarity – then many blend stages would likely be muddled or incoherent, causing the text to fall below q .

Hence, it is reasonable to suspect that the high-quality subgraph might contain distinct islands that cannot reach one another without temporarily leaving $G_x^{\geq q}$. In fact, when humans write – one character at a time – they invariably pass through numerous low-quality states (partial words, half-formed sentences) before arriving at any one of the various ways of saying something of quality. Local edit operators, such as those that insert or delete single tokens or small chunks of text, face a similar risk: even a small disruption can degrade quality if the threshold is strict.

That said, irreducibility might still be recovered if we loosen our assumptions. For instance, we might allow momentary dips in quality during transitions so long as the process does not get stuck below q ; or we could permit larger, more context-aware edits that can leap more cleanly between stylistic domains. In practice, these motivations led to the development of the **Document2StepMutator**, which aims to ensure that modifications are localized enough to avoid substantial quality degradation, yet also sufficiently broad to permit meaningful jumps. This design tries to strike a balance between remaining near high-quality states most of the time and retaining enough flexibility to move across different regions of the text space – ideally preventing the formation of disconnected islands of high-quality text.

Why do LLMs Struggle to Verify? While humans intuitively find verification easier than generation, this asymmetry may actually *reverse* for LLMs due to their probabilistic architecture and training paradigms. The core tension arises from LLMs’ design as next-token predictors [31], which optimizes them for fluency over factual accuracy or logical rigor [21, 181]. Though techniques like chain-of-thought prompting [314] can simulate self-checking, the models remain fundamentally tuned to generate plausible continuations – not to verify them.

Compounding this, LLMs lack exposure to the iterative critique processes that shape human judgment. Trained on polished outputs [75], they rarely encounter explicit revisions (e.g., drafts with margin notes like "this plot point contradicts Chapter 3") that teach cause-effect relationships between quality and text structure [289]. Consequently, their "critiques" often reduce to surface-level heuristics (e.g., associating complex syntax with professionalism) rather than principled reasoning.

Whether verification is inherently harder for LLMs may hinge on whether quality is reducible

to likelihood. If not, their adeptness at generating fluent text may paradoxically hamper verification, as polished outputs mask subtle shortcomings [21], creating a hall-of-mirrors effect where plausibility is mistaken for truth.

8.6 Conclusion

Our findings reveal that watermark removal via random-walk attacks is far less certain than theoretical work suggests. Slow mixing and imperfect quality verification create significant real-world barriers. These insights invite deeper investigation: evolving watermark schemes could exploit the difficulty of consistent, high-quality edits, while attackers must grapple with the costs and risks of large-scale text manipulation. Our study also highlights the need for quality measures that align with human judgment, not surface features. Addressing these challenges – mixing speed, oracle reliability, and quality standards – will ensure watermarking remains viable against sophisticated attacks.

8.7 Appendix

8.7.1 Formal Definitions

In this section, we provide formal definitions of objects mentioned in Section 8.2 and elaborate on some definitions. As with the background section, most of these are directly from [341]. Let us begin by providing formal definitions of objects mentioned in Section 8.2.

Definition 8.7.1 (ϵ_{pert} -Preserving Perturbation Oracle, [341], Definition 6). Let $\mathbf{P} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Y}$ be a randomized oracle that, given (x, y) , outputs a new response y' . The oracle $\mathbf{P}$ is said to be ϵ_{pert} -preserving if for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,

$$\Pr\left[\mathbf{Q}(x, \mathbf{P}(x, y)) \geq \mathbf{Q}(x, y)\right] \geq \epsilon_{\text{pert}}.$$

Definition 8.7.2 ([341], Definition 8). Let $G = (V, E)$ be a weighted directed graph, and $\vec{P}$ be the transition matrix of G . We say that $\vec{\pi} \in \mathbb{R}^n$ is a stationary distribution for $\vec{P}$ if: $\vec{P}^\top \cdot \vec{\pi} = \vec{\pi}$.

Definition 8.7.3. A weighted directed graph $G = (\mathcal{V}, \mathcal{E})$ is **irreducible** if for any pair of vertices $u, v \in \mathcal{V}$, there exists a directed path from u to v with non-zero weight. In other words, there exists some $t \geq 1$ such that $\vec{P}^t(i, j) > 0$.

Definition 8.7.4. A weighted directed graph $G = (\mathcal{V}, \mathcal{E})$ is **aperiodic** if the greatest common divisor of the lengths of all directed cycles in G is 1.

Let us now formally define the (hierarchically ordered) graph representations of $\mathbf{P}$ based on a prompt $x \in \mathcal{X}$ and the quality threshold $q \in [0, 1]$.

Definition 8.7.5 ([341], Definition 7). Fix an arbitrary prompt $x \in \mathcal{X}$ and consider the graph $\mathbf{G}_x = (\mathcal{V}_x, \mathcal{E}_x)$ whose vertex set is the output space of $\mathbf{M}$ (i.e., $\mathcal{V}_x = \mathcal{Y}$) and whose edge set $\mathcal{E}_x$ consists of all pairs (y, y') such that

$$\Pr[y' = \mathbf{P}(x, y)] > 0.$$

We assign weights $w : \mathcal{E}_x \rightarrow [0, 1]$ to the edges by defining

$$w(y, y') = \Pr[y' = \mathbf{P}(x, y)].$$

Note that while the vertices of the graph are determined by the prompt $x \in \mathcal{X}$ and the watermarking model $\mathbf{M}$, the edges and their weights are determined solely by $\mathbf{P}$. Let us now incorporate quality into the graph representation. Let $\mathbf{G}_x^{\geq q}$ be the subgraph of $\mathbf{G}_x$ given by

$$\mathcal{V}_x^{\geq q} = \{y \in \mathcal{Y} \mid \mathbf{Q}(x, y) \geq q\},$$

$$\mathcal{E}_x^{\geq q} = \{(y, y') \in \mathcal{Y} \times \mathcal{Y} \mid \mathbf{Q}(x, y) \geq q, \mathbf{Q}(x, y') \geq q, \Pr[y' = \mathbf{P}(x, y)] > 0\},$$

Notice that we can carry the same weight assignment to this subgraph. Iteratively applying $\mathbf{P}$ on this graph and rejecting low-quality mutations produces a random walk where

$$\vec{P}_{(y,y')} = \Pr[y' = \mathbf{P}(x, y)].$$

Before presenting the WITS impossibility result, we formally define watermarking schemes and related notions.

Definition 8.7.6 ([341], Definition 3). Let $\mathcal{M} = \{\mathbf{M}_i : X \rightarrow Y\}$ be a class of generative models with key space K . A secret-key watermarking scheme for $\mathcal{M}$ consists of two efficient algorithms:

- **Watermark**($\mathbf{M}$): A randomized algorithm that, given a model $\mathbf{M} \in \mathcal{M}$, outputs a secret key $k \in K$ and a corresponding watermarked model $\mathbf{M}_k : X \rightarrow Y$.
- **Detect** $_k(x, y)$: A deterministic algorithm that, given a secret key $k \in K$, a prompt $x \in X$, and an output $y \in Y$, returns a bit $b \in \{0, 1\}$ indicating whether the watermark is present ($b = 1$) or absent ($b = 0$).

We now define the false-positive rate ϵ_{pos} of a watermarking scheme. Notice that if we sample $y \stackrel{\$}{\leftarrow} \vec{\pi}$, we expect that y is unwatermarked with probability ϵ_{pos} . Since for every good watermarking scheme we need ϵ_{pos} to be very small, this implies that outputs sampled from $\vec{\pi}$ will be unwatermarked with high probability, i.e. with probability approximately $1 - \epsilon_{\text{pos}}$.

Definition 8.7.7 ([341], Definition 4). A watermarking scheme Π has a false positive ϵ_{pos} -rate if, for every model $M \in \{\mathbf{M}_i : X \rightarrow Y\}$, for every prompt $x \in X$, and for every output $y \in Y$,

$$\Pr[\text{Detect}_k(x, y) = 1] \leq \epsilon_{\text{pos}}.$$

Let us now define what it means for an adversary to break a watermarking scheme Π . Notice that one might consider weaker notions of breaking a watermarking scheme, but they will be implied by the WITS result.

Definition 8.7.8 ([341], Definition 5). Let $\Pi = (\text{Watermark}, \text{Detect})$ be a watermarking scheme for a class of generative models $\mathcal{M} = \{\mathbf{M}_i : X \rightarrow Y\}$ with associated quality function $\mathbf{Q} : X \times Y \rightarrow [0, 1]$. We say that an adversary A ϵ -breaks Π if for every $M \in \mathcal{M}$, for every prompt

$x \in X$, we have:

$$\Pr [\text{Detect}_k(x, y') = 0 \text{ and } \mathbf{Q}(x, y') \geq \mathbf{Q}(x, y) : y \stackrel{\$}{\leftarrow} \mathbf{M}_k(x), y' \stackrel{\$}{\leftarrow} A(x, y)] \geq \epsilon$$

where the probability is taken over $(k, \mathbf{M}_k)$ output by $\text{Watermark}(M)$ and the random coins of A .

We now introduce an additional technical definition that enables the attacker to trade off between output quality and attack success probability. Let $v \in [0, 100]$ denote the desired quality percentile. In other words, the attacker aims to produce an unwatermarked output whose quality falls within the top v -th percentile among all responses generated by $\mathbf{M}$ on a given prompt x . To formalize this, define the set of achievable quality scores as

$$\mathbf{Q}_{M,x} = \left\{ q : \Pr[\mathbf{Q}(x, \mathbf{M}_k(x)) = q : (k, \mathbf{M}_k) \stackrel{\$}{\leftarrow} \text{Watermark}(M)] > 0 \right\}$$

and let $q_{\mathbf{M},x}$ denote the v -th percentile of $\mathbf{Q}_{M,x}$. We then define the overall minimum quality threshold as

$$q_{\min} = \min_{\mathbf{M} \in \mathcal{M}, x \in \mathcal{X}} \{q_{\mathbf{M},x}\}.$$

We now state the WITS impossibility result.

Theorem 1 ([341], Theorem 6). Let $\Pi = (\text{Watermark}, \text{Detect})$ be a watermarking scheme for a class of generative models $\mathcal{M} = \{\mathbf{M}_i : \mathcal{X} \rightarrow \mathcal{Y}\}$ with an associated quality function $\mathbf{Q} : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$. Let $\mathbf{P} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Y}$ be a perturbation oracle (defined over the same prompt space $\mathcal{X}$ and output space $\mathcal{Y}$ as the class $\mathcal{M}$) with the same associated quality function $\mathbf{Q} : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$ as Π . For every non-watermarked model $\mathbf{M} \in \mathcal{M}$, for every prompt $x \in \mathcal{X}$, for every quality $q \in q_{\min}$, let $\vec{\pi}_{x,q}$ be the unique stationary distribution of the transition matrix $\vec{P}_{x,q}$ of $G_x^{\geq q}$. Let $n_{x,q} = |\mathcal{V}_x^{\geq q}|$, $\pi_{\min}^{(x,q)} = \min\{\vec{\pi}_{x,q}(1), \dots, \vec{\pi}_{x,q}(n_{x,q})\}$ and g be the second largest eigenvalue of $\vec{P}_{x,q}$ in terms of absolute value. Let $t_{\text{err}} > 0$ be a tunable parameter. Let $t_{x,q}$ be the ϵ_{dist} -mixing time of $\vec{P}_{x,q}$, defined as follows:

$$t_{x,q} = \omega \left(\frac{1}{1-g} \cdot \log \left(\frac{1}{\pi_{\min}^{(x,q)} \cdot \epsilon_{\text{dist}}} \right) \right)$$

Assume the following holds:

1. The watermarking scheme Π has a false positive ϵ_{pos} -rate;
2. The perturbation oracle $\mathbf{P}$ is ϵ_{pert} -preserving;
3. For every non-watermarked model $\mathbf{M} \in \mathcal{M}$, for every prompt $x \in \mathcal{X}$, for every quality $q \in [q_{\min}, 1]$, the q -quality x -prompt graph representation $\mathbf{G}_x^{\geq q}$ of $\mathbf{P}$ is irreducible and aperiodic.

Then, there exists an oracle-aided universal adversary $A^{\mathbf{P}(\cdot, \cdot), \mathbf{Q}(\cdot, \cdot)}$ that ϵ -breaks Π by submitting at most t queries to $\mathbf{P}$ where

$$\epsilon = \left(1 - \frac{v}{100}\right) (1 - \epsilon_{\text{pos}})(1 - \epsilon_{\text{dist}}) \left(1 - \sum_{k=0}^{t-t_{\text{err}}-1} \binom{t}{k} (\epsilon_{\text{pert}})^k (1 - \epsilon_{\text{pert}})^{t-k}\right),$$

and

$$t = \max_{x \in \mathcal{X}, q \in [q_{\min}, 1]} \{t_{x,q}\} + t_{\text{err}}.$$

By carefully tuning the parameter t_{err} and running the attack long enough so that ϵ_{dist} becomes negligibly small, the adversary can achieve a success probability close to

$$\left(1 - \frac{v}{100}\right) (1 - \epsilon_{\text{pos}}).$$

For example, targeting the median quality output (i.e., setting $v = 50$) restricts the adversary's success probability to roughly half of the maximum achievable rate.

8.7.2 Evaluation Setup

8.7.3 Watermark Details

For KGW and SIR, we use the implementations contained within the `MarkLLM` package² [233] with their default configurations. For Adaptive, we used the author’s implementation³ and due to initially poor results, experimented heavily with different configurations to find one that best balanced initial quality and detectability for `Llama-3.1-70B-Instruct`. The three tunable parameters we explored were `alpha`, which thresholds the amount of token entropy required to watermark it; `delta`, which controls the strength of boosting for watermarked tokens; and `delta_0`, which is the strength for watermarking the first $M = 50$ tokens, which are always watermarked. Our analysis lead us to use `alpha = 2.0`, `delta = 1.5`, `delta_0 = 1.0`. Despite this extensive search, we still encountered intermittent issues with controlling for generation length. Since all texts were capped at a maximum of 1024 tokens due to fixed input sizes for various embedding models, some Adaptive responses were truncated mid-sentence, contributing to their unusually high number of grammatical errors as seen in Table 8.5.

By plotting the distributions for each quality metric in Figures 8.1 through 8.4, we noticed that Adaptive and SIR were vulnerable to producing highly distorted text with numerous quality issues. For example, a single Adaptive generation contained over 250 grammatical issues, largely due to inexplicable letter case alterations (e.g. “Over ThE nexT FEw dAYs, maggie partICIpaTed EnThusiasticALly I-n All ThE acTivities OffeRed aT WIllOW crEeEk...”). We did not regenerate bad responses because the distortions were a natural consequence of the watermarking algorithm itself, and regenerating them would obscure an important challenge to their real-world use. If the algorithm produces highly distorted text in some cases, then an attack is actually more likely to *repair* quality, rather than merely preserving it. At least some cases in our study fit this profile and the attack should be fairly credited even if it generally does not work for texts that start at a higher standard of quality.

²<https://github.com/THU-BPM/MarkLLM>

³<https://github.com/yepengliu/adaptive-text-watermark>

InternLM Quality Distribution by Model

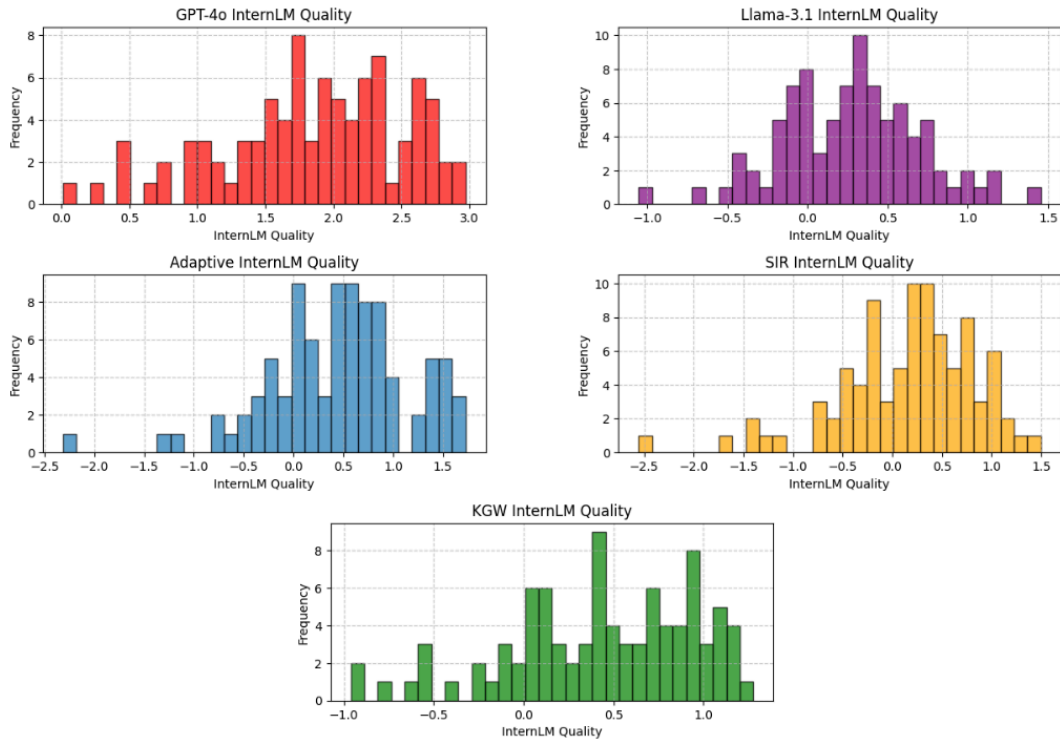


Figure 8.1: InternLM Quality Distribution by Watermarking Scheme W

Perplexity Distribution by Model

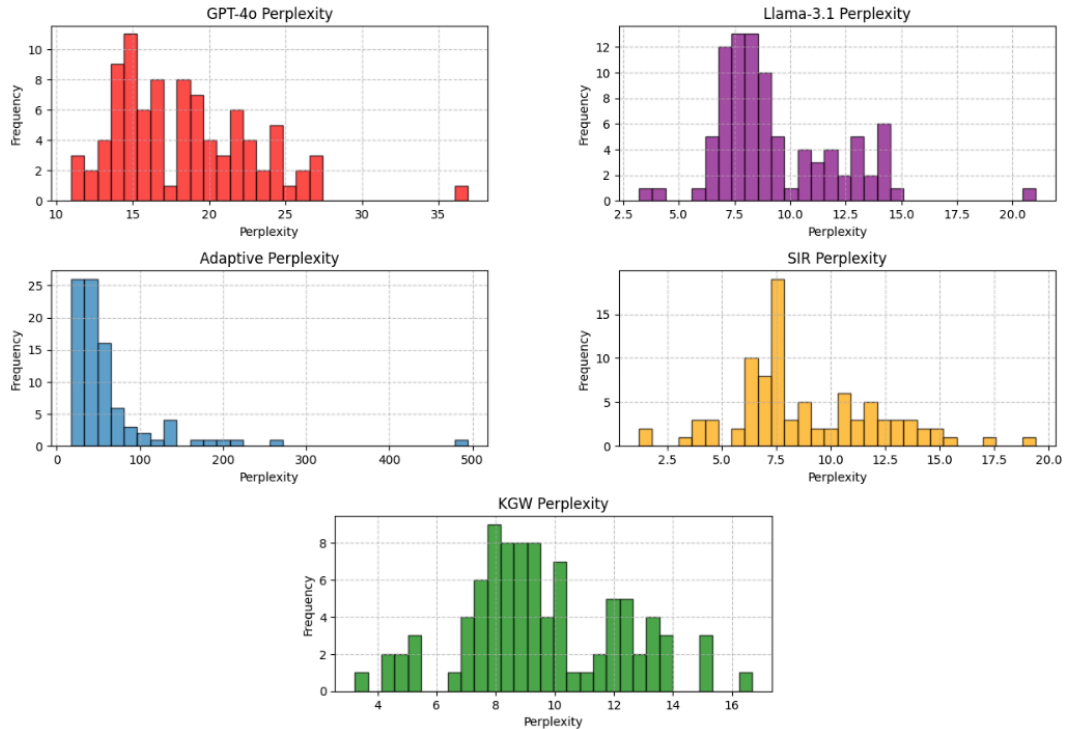


Figure 8.2: Perplexity Distribution by Watermarking Scheme W

Unique Bigrams Distribution by Model

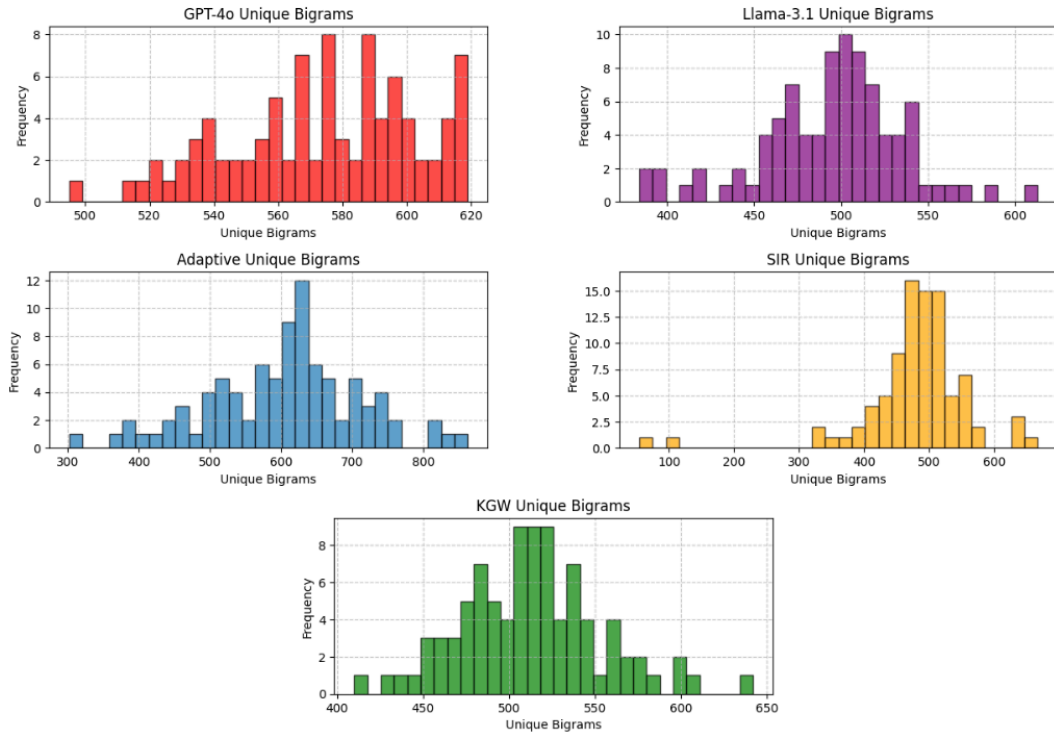


Figure 8.3: Unique Bigrams Distribution by Watermarking Scheme W

Grammar Errors Distribution by Model

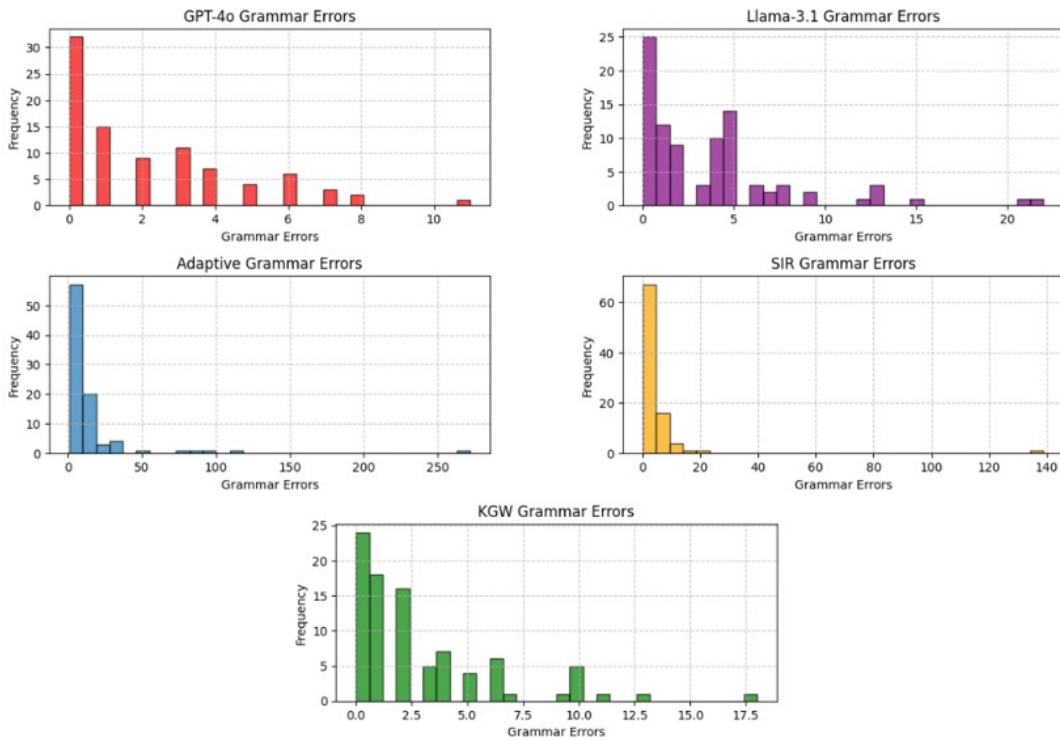


Figure 8.4: Grammar Errors Distribution by Watermarking Scheme W

8.7.4 Entropy-Controlled Prompt Dataset

To systematically evaluate the impact of response entropy on watermark robustness, we curated a dataset featuring increasingly specific prompts across three domains: **creative writing**, **education**, and **journalism**. For each domain, we start with a broad, high-entropy prompt and progressively add constraints to reduce entropy. Below, we illustrate this progression with representative prompts at entropy level 1 (least constrained), 5, and 10 (most constrained).

Entropy Level	Prompt
Creative Writing	
1	Write a 500-word story.
5	Write a 500-word story about Evan, an American tourist, who falls for Emilie, a barista, during a spring festival in Paris.
10	Write a 500-word story about Evan, an American tourist, who falls for Emilie, a barista, during a spring festival in Paris. They bond over their love for Claude Monet's 'Impression, Sunrise' and the Hotel de Sully's architecture, leading to walks along the Seine. Their connection deepens amid shared laughter and explorations of Le Marais. As the festival lights dance on the river, Evan shares his feelings with Emilie under the starlit sky, promising to cherish the moments they've shared.
Education	
1	Write a 500-word essay about the importance of space exploration.
5	Write a 500-word essay about the importance of space exploration, its role in advancing human knowledge, and its potential to address global challenges like climate change and resource scarcity, with a focus on technologies developed for space missions.
10	Write a 500-word essay about the importance of space exploration, its role in advancing human knowledge, and its potential to address global challenges like climate change and resource scarcity, with a focus on technologies developed for space missions, their applications on Earth, the possibility of colonizing other planets like Mars, the ethical considerations of interplanetary exploration, and the cultural significance of humanity becoming an interstellar species.
Journalism	
1	Write a 500-word news article.
5	Write a 500-word news article about a global climate summit where world leaders are discussing strategies to combat climate change, with a focus on renewable energy investments and carbon reduction targets, highlighting a groundbreaking agreement between the US and China.
10	Write a 500-word news article about a global climate summit where world leaders are discussing strategies to combat climate change, with a focus on renewable energy investments and carbon reduction targets, highlighting a groundbreaking agreement between the US and China, featuring perspectives from small island nations affected by rising sea levels, addressing protests outside the summit calling for stronger climate justice measures, covering a controversial speech by a major oil industry representative, analyzing the summit's key outcomes and challenges, and placing it in the broader context of international efforts to achieve net-zero emissions by 2050.

Table 8.4: Representative entropy-controlled prompts across three domains: creative writing, education, and journalism. Entropy increases by adding specificity, progressively constraining the response space.

8.7.5 Dataset Statistics

	Unwatermarked		Watermarked		
	GPT-4o	Llama-3.1	Adaptive	SIR	KGW
Mean Watermarked Score (μ_w)	–	–	99.27	0.28	5.32
Mean Unwatermarked Score (μ_{uw})	–	–	49.43	0.08	-0.83
Unwatermarked Standard Deviation (σ_{uw})	–	–	3.37	0.07	1.05
Breakpoint (Score $\leq \mu_{uw} + 2\sigma_{uw}$)	–	–	56.16	0.21	1.27
Quality Score	1.85	0.27	0.45	0.16	0.43
Perplexity	18.39	9.38	63.32	8.87	9.56
Grammar Errors	2.20	3.69	16.21	4.24	2.86
Unique Bigrams Diversity	574.06	494.59	603.64	479.90	512.44
Mean Word Count	637.93	633.00	646.62	675.47	666.84
Generation Time (s)	15.24	274.27	671.75	335.12	292.24
Detection Time (s)	–	–	240.77	5.78	0.15

Table 8.5: Summary statistics for unwatermarked and watermarked text across different watermarking schemes, highlighting detection scores, automatic quality metrics, and runtime statistics.

8.7.6 Perturbation Oracle Details

The perturbation oracles $\mathbf{P}$ define the mechanism by which adversarial modifications are applied to watermarked text. These oracles generate perturbations of varying granularity, from token-level edits to full-document paraphrasing, enabling a systematic analysis of their impact on watermark robustness. Since prior work, including [341], has not accounted for how different perturbation strategies affect attack success, we explore a diverse set of perturbation oracles to quantify their relative effectiveness.

- **WordMutator**: Randomly replaces individual tokens by masking and filling them using RoBERTa [188].
- **EntropyWordMutator**: Similar to WordMutator, but uses GPT-Neo-2.7B [25] to target high-entropy tokens for replacement as they are most likely to carry watermark signals.
- **SpanMutator**: Randomly masks six contiguous tokens at a time and fills them using T5-XL [251]. This is the only text-based perturbation oracle used in the WITS attack [341].
- **SentenceMutator**: Randomly selects a sentence and paraphrases it creatively using Llama-3.1-8B [78], introducing higher-level semantic shifts.
- **DocumentMutator**: Uses the DIPPER paraphrase model [153] to paraphrase multiple sections of the document simultaneously.
- **Document1StepMutator**: Re-generates the entire document from scratch using Llama-3.1-8B, producing the most extreme form of perturbation while preserving meaning, quality, and formatting.
- **Document2StepMutator**: Performs a two-step transformation, first selecting a random sentence and paraphrasing it creatively with Llama-3.1-8B, then performs a global consistency editing to ensure that the remaining text is consistent with the edited sentence.

These perturbation oracles serve two key purposes in our study: (1) they enable us to analyze how the size of the perturbation affects movement within the perturbation graph $\mathbf{G}$; and (2) they allow us to determine whether specific perturbation oracles are more effective at breaking watermarks. By systematically evaluating these oracles, we aim to establish whether certain perturbation strategies inherently favor watermark removal and whether prior work has underestimated their im-

pact on attack success.

To ensure sufficient opportunity for watermark removal, we allow a large number of perturbation steps, proportional to the average number of words edited per step. For example, WordMutator is given 1000 steps, while DocumentMutator is given 100. Additionally, we note that each perturbation oracle was carefully calibrated to balance subtle modifications with sufficient impact on watermark signals, ensuring reproducibility by fixing random elements such as token selection and sampling temperature. Table 8.6 reveals a clear trade-off: while fine-grained oracles tend to preserve fluency, coarse-grained methods introduce larger variations – a difference that is partly mitigated by the consistency editing in the Document2StepMutator.

P	Steps	Edits	PPL ↓	Gram Err ↓	Approval ↑	Blocked ↓	QScore ↑	Time (s) ↓
Word	1000	1.8	40.4	10.2	0.80	0	-0.0688	0.10
EntropyWord	1000	1.1	16.8	9.2	0.82	0	-0.0949	0.28
Span	250	8.7	27.2	7.8	0.67	0	-0.0746	0.77
Sentence	150	31.3	21.0	4.6	0.74	0	0.2065	0.94
Document	100	216.0	11.2	9.1	0.36	0.12	-0.3151	213.12
Document1Step	100	138.2	10.6	2.1	0.42	0.12	0.0536	29.61
Document2Step	100	105.2	14.9	5.2	0.54	0.03	0.2089	34.78

Table 8.6: Performance metrics for each perturbation oracle. The columns report the number of attack steps, average edits per step, average text perplexity (PPL), average number of grammar errors, average **Q** approval rate, average rate at which **Q** blocks every perturbation for a given prompt, average InternLM quality score (QScore), and average runtime per perturbation step in seconds. Emboldened values denote the best performance per metric.

8.7.7 Distinguisher Prompt

Designing a distinguisher requires careful consideration, as failures can be difficult to attribute to either implementation flaws or inherent limitations due to sufficient mixing. To address this challenge, we dedicated significant effort to prompt engineering, developing 12 different distinguishers and systematically benchmarking their performance. Somewhat unexpectedly, one of the simplest approaches proved to be the most effective, and we describe it below.

An important implementation detail is ensuring that our distinguisher does not pass tests due to random chance (as there are only 2 options it must choose between). Each test consists, on average, of 10 periodically sampled texts throughout the random walk. The test is considered a failure if even one sample fails, ensuring that the distinguisher can accurately deduce the origin throughout

the entire attack. Additionally, we find that when the distinguisher fails, it almost always selects response A as the origin. We believe this to be due to positional bias. To combat this, every sample is distinguished as a best-of-2, where the second trial swaps response A and response B. The sample is considered a failure if either trial fails, minimizing the chance that the distinguisher succeeds by guessing. We remark that in every instance where the distinguisher failed, the best-of-2 was a tie, and zero samples (out of 57650) were incorrectly distinguished in both trials. This suggests the model was sometimes unable to overcome positional bias, but was never directly wrong. In these cases, we defer the final verdict to a stronger model, ultimately resulting in 100% accuracy.

The following Python code defines the main method of `SimpleDistinguisher`, implemented with `guidance` [193] to support constrained decoding. This function is designed to analyze response similarity while leveraging a predefined persona. The persona guides the AI model in determining whether a given perturbed response (P) originated from one of two original responses (A or B).

```
1 persona = \
2 """
3 You are an expert in analyzing the similarity of responses.
4 You specialize in detecting whether one response has been derived from
   another by carefully analyzing the content and structure of the text.
5 Your expertise in linguistic analysis allows you to distinguish which
   responses are the most closely related.
6 Your goal is to provide a clear, concise, and accurate assessment of the
   provided instructions.
7 """
8
9 @guidance
10 def distinguish(lm, persona, **kwargs):
11     if persona:
12         with system():
13             lm += f"{persona}"
14         with user():
15             lm += f"""
16             ### Task Description:
```

```

17         1. Read the following 3 responses. They are labelled "A", "B", and
        "P".
18         2. "A" and "B" are original responses, and "P" was generated by
        perturbing one of the original responses.
19         3. Identify which of the original responses the perturbed story was
        derived from.
20
21         Original response A:
22         {kwargs["A"]}
23
24         Original response B:
25         {kwargs["B"]}
26
27         Perturbed response P:
28         {kwargs["P"]}
29         """
30     with assistant():
31         lm += f"""
32         I believe the perturbed response P was derived from original
        response {select(["A", "B"], name="choice")}.
33         """
34     return lm

```

Listing 8.1: Distinguisher Prompt Template

8.7.8 A Challenging Distinguisher Example

As an example of the data used with the prompt above, we present an example that was particularly challenging for our distinguishers. After 108 sentence-level perturbations, GPT-4o was unable to accurately distinguish the origin. With around 30 seconds of reasoning, o3-mini-high correctly distinguished the origin, but had to correct itself while reasoning. We remark that the final section on NASA’s Artemis program makes distinguishing this example trivial for humans, suggesting that our distinguishers are significantly weaker than humans. The perturbed text, along with the two original responses, are provided below with some key phrases in bold.

Perturbed Text (GPT-4o Failed to Distinguish)

enturing into space is a groundbreaking endeavor that unlocks a multitude of benefits, extending far beyond the realms of scientific discovery and territorial growth. Space exploration, frequently overlooked, is a catalyst for scientific progress, driving the development of pioneering technologies and addressing humanity's most pressing challenges directly, making it a pursuit of paramount importance that warrants greater acknowledgment and support. This essay examines the importance of space exploration, its potential to broaden our understanding, and its ability to contribute to resolving critical global challenges like environmental decay and resource exhaustion. Understanding the cosmos is vital, as it allows us to grasp the intricate mechanisms **governing the universe and our place within it**, ultimately expanding our comprehension of reality itself. Exploring the vastness of the universe reveals a profound comprehension of the fundamental laws that shape reality, the origin of life, and the intricate chronology of cosmic evolution that has spanned eons of time. Delving deeper into our environment not only quenches our innate desire for knowledge but also empowers us to make more informed choices about the planet's destiny, thereby shaping our relationship with the world that surrounds us. The pursuit of space exploration has far-reaching consequences, resulting in numerous groundbreaking discoveries that cumulatively contribute to a significant improvement in global well-being, manifesting in a multitude of tangible advantages. The rapid evolution of technology, encompassing satellite communication, GPS, and medical imaging, has significantly influenced our daily routines, work, and relationships, transforming the way we interact and live our lives. Advances in technology have not only bridged the world but have also led to better health outcomes worldwide, significantly impacting our daily lives and perceptions. Beyond its contributions to science and technology, space exploration provides **a distinctive vantage point for understanding the Earth and its interconnected systems**. Viewing our planet from space offers a comprehensive understanding of the interconnectedness of Earth's atmospheric, oceanic, and terrestrial systems, showcasing a cohesive entity that surpasses its individual components in complexity and unity.

Understanding the effects of human actions on the environment is crucial for tackling pressing global issues, such as climate change, which is becoming more apparent with each passing day. Satellite imagery has been instrumental in tracking climate shifts, monitoring the growth of our oceans, and forecasting extreme weather events, all of which are crucial for comprehending the intricate dynamics of our planet's ever-changing environment. Creating a resilient and lasting future demands a sophisticated understanding of climate change's multifaceted impacts and the strategic application of targeted solutions to minimize its effects with accuracy and efficiency. One of the most significant benefits of space exploration is its potential to alleviate the consequences of dwindling resources on Earth. As the global population grows, the need for fundamental resources like food, water, and power is escalating, necessitating a significant boost in production to meet the rising demand. Exploring space can reveal previously unknown sources of essential materials, including minerals and water, which can be exploited to satisfy the growing need for these vital resources. **By 2024, NASA's Artemis mission seeks to create a self-sustaining lunar community**, ultimately paving the way for a permanent human settlement on the Moon that enables ongoing exploration and habitation. This expedition may reveal new helium-3 deposits, a scarce isotope that could serve as a fuel for nuclear fusion reactions, offering a potential source of power. Ultimately, space exploration is crucial for expanding our understanding of the universe, fostering groundbreaking technological advancements, and tackling pressing global issues such as environmental degradation and resource depletion. Venturing into the unknown reaches of space not only expands our comprehension of the cosmos, but also paves the way for groundbreaking discoveries that, in turn, significantly influence our daily lives and overall well-being. Satellite imaging provides a distinct viewpoint, allowing us to comprehend the intricacies of the obstacles we encounter and tackle them with greater success. As we venture deeper into the cosmos, we may stumble upon innovative answers to humanity's most critical challenges, thereby ensuring a thriving and sustainable world for future generations to

inherit.

(V)

Original Response A

space exploration is an essential endeavor for humanity, offering numerous benefits that extend beyond the realm of scientific discovery. While some may view it as a luxury or a frivolous pursuit, the reality is that space exploration plays a vital role in advancing human knowledge, driving technological innovation, and addressing some of the most pressing global challenges we face today. In this essay, we will explore the importance of space exploration, its contributions to human understanding, and its potential to help mitigate global crises like climate change and resource scarcity. One of the primary reasons space exploration is essential is that it allows us to expand our **understanding of the universe and our place within it**. By exploring the cosmos, we gain valuable insights into the fundamental laws of physics, the origins of life, and the evolution of the universe. This knowledge not only satisfies our innate curiosity but also informs our understanding of the world around us, enabling us to make more informed decisions about our planet and its resources. Furthermore, space exploration drives technological innovation, which has numerous spin-off benefits for society. The development of new technologies, such as satellite communications, GPS, and medical imaging, has transformed the way we live, work, and interact with one another. These innovations have also improved our daily lives, from enabling global connectivity to enhancing healthcare outcomes. In addition to its scientific and technological benefits, space exploration also offers a **unique perspective on the Earth and its systems**. By studying the Earth from space, we gain a deeper understanding of the complex interactions between our atmosphere, oceans, and landmasses. This knowledge is essential for addressing global challenges like climate change, where the impacts of human activity on the planet are becoming increasingly evident. Space-based observations have played a critical role in monitoring climate change, tracking sea-level rise, and predicting weather patterns. These insights are essential for de-

veloping effective strategies to mitigate the effects of climate change and ensure a sustainable future. Another significant benefit of space exploration is its potential to address resource scarcity. As the global population continues to grow, the demand for resources like food, water, and energy is increasing. Space exploration offers a unique opportunity to identify new sources of resources, such as minerals and water, that can help meet this demand. For example, **NASA's Artemis program aims to return humans to the Moon by 2024**, with the long-term goal of establishing a sustainable presence on the lunar surface. This mission could potentially unlock new resources, such as helium-3, a rare isotope that could be used as fuel for nuclear fusion. In conclusion, space exploration is essential for advancing human knowledge, driving technological innovation, and addressing global challenges like climate change and resource scarcity. By exploring the cosmos, we gain valuable insights into the universe and our place within it, while also developing new technologies that improve our daily lives. Furthermore, space-based observations offer a unique perspective on the Earth and its systems, enabling us to better understand and address the complex challenges we face. As we continue to explore the vast expanse of space, we may uncover new solutions to some of humanity's most pressing problems, ensuring a sustainable and prosperous future for generations to come.

(S)

Original Response B

pace exploration is an essential endeavor for humanity, offering numerous benefits that extend beyond the realm of scientific discovery. It plays a vital role in advancing human knowledge, driving technological innovations, and addressing pressing global challenges such as climate change and resource scarcity. In this essay, we will explore the significance of space exploration and its potential to shape the future of our planet and beyond. The pursuit of space exploration is often viewed as a costly and ambitious endeavor, but it is essential to recognize the significant contributions it makes to our **understanding of the universe and the world we inhabit**. By venturing into space,

we gain insights into the fundamental laws of physics, the origins of life, and the evolution of the cosmos. These discoveries not only expand our scientific knowledge but also inspire new generations of scientists, engineers, and innovators. Furthermore, space exploration has led to numerous technological innovations that have transformed various aspects of our daily lives. From the development of GPS and telecommunications to medical imaging and weather forecasting, the spin-off benefits of space exploration have been substantial. These innovations have improved the quality of life for millions of people around the world and have also generated significant economic benefits. In addition to its scientific and technological benefits, space exploration also offers a **unique opportunity to address pressing global challenges**. For instance, the study of Earth from space provides critical insights into the health of our planet and the impacts of climate change. Satellite imaging and remote sensing technologies have enabled scientists to monitor deforestation, track ocean currents, and detect changes in global temperature patterns. This information is essential for developing effective strategies to mitigate the effects of climate change and promote sustainable development. Another significant benefit of space exploration is its potential to provide new resources and opportunities for economic growth. As the world's population continues to grow, the demand for resources such as food, water, and energy will increase. Space exploration offers a way to address this challenge by accessing new sources of resources, such as asteroid mining and lunar helium-3 extraction. These resources could provide a clean and sustainable source of energy, reducing our reliance on fossil fuels and mitigating the impacts of climate change. Finally, space exploration offers a unique opportunity for international cooperation and diplomacy. In an era marked by increasing global tensions and conflict, space exploration provides a shared goal that can bring nations together. Collaborative efforts such as the International Space Station and the **Artemis program** have demonstrated the potential for space exploration to foster global cooperation and understanding. In conclusion, space exploration is essential for advancing human knowledge, driving technological innovations, and addressing pressing global challenges. Its significance extends beyond

the realm of scientific discovery, offering numerous benefits that have the potential to shape the future of our planet and beyond. As we continue to explore the vastness of space, we must recognize the importance of investing in this endeavor and working together to address the challenges that lie ahead. By doing so, we can ensure that the benefits of space exploration are shared by all and that the next generation of scientists, engineers, and innovators is inspired to reach for the stars.

(S)

8.7.9 Extended Distinguisher Study

In addition to the main **RQ1** result, we designed an even more challenging evaluation setting to test whether sufficient mixing could obscure the lineage of perturbed texts. Specifically, we focus on the strongest **P**, `SentenceMutator`, as it previously demonstrated the highest capacity to evade detection by `Llama-3.1-70B`. To amplify its effect, we increase the perturbation budget from 150 to 500 steps, allowing the random walk significantly more opportunities to approach the stationary distribution.

Additionally, we constrain the attack to texts generated from the lowest-entropy prompts, ensuring that candidate parent texts are highly similar to one another. This combination of (1) the strongest perturbation oracle, (2) an extended attack budget, and (3) a highly confounded candidate pool creates the most difficult setting for lineage attribution. If mixing is truly effective under these conditions, we would expect distinguishability to approach random chance.

We find that although the task was more challenging, with more failures on average, `o3-mini-high` still had no issues in distinguishing the origin in each test.

8.7.10 Breakdown by Domain and Entropy

We find domain to be significant in distinguishability, but surprisingly, not entropy.

P Oracle	Steps	Tests	Llama-3.1-70B	GPT-4o	o3-mini-high
Sentence	500	54	13	3	0
Cumulative Distinguished (%)			75.9	94.4	100

Table 8.7: Summary of failed distinguisher tests on the most challenging settings. Classification is first performed by Llama-3.1-70B, followed by GPT-4o on its failures, then o3-mini-high on any remaining cases. The overall 100% success rate indicates that the attacked texts never lose memory of their starting points, contradicting **KA1** and suggesting that a stationary distribution is not reached in practice.

Domain	Failed Distinguishes (Main)	Failed Distinguishes (Challenge)
Journalism	6/1458	0/18
Creative Writing	7/1560	6/18
Education	40/1537	7/18

Table 8.8: Domain distribution for tests which Llama-3.1-70B failed to distinguish.

Entropy	Failed Distinguishes (Main)	Failed Distinguishes (Challenge)
1	7/462	N/A
2	4/457	N/A
3	8/468	N/A
4	9/462	N/A
5	3/450	N/A
6	1/468	N/A
7	6/450	N/A
8	2/438	N/A
9	5/456	N/A
10	8/444	13/54

Table 8.9: Entropy distribution for tests which Llama-3.1-70B failed to distinguish.

8.7.11 Quality Oracles

8.7.11.1 Oracle Details

The quality oracles determine whether the perturbations introduced by various **P** preserve the original texts quality. Each oracle operates by querying an LLM with a prompt and some continuation text using different strategies to assess preservation of meaning, fluency, and coherence. The quality decision is based on whether the mutated text is judged to be as good as or better than the original. All oracle queries include the original prompt to provide context for evaluation.

We implement and evaluate eight distinct quality oracles using guidance [193] to support constrained decoding for ranking, scoring, and preference based assessments.

- **RankOracle:** **Q** is prompted to rank the two responses in terms of preference, and the order of texts is then reversed in a second query. If the mutated text is preferred in both cases, quality is considered preserved.
- **SoloOracle:** **Q** is prompted twice, independently grading each text on a numerical scale. If the mutated text receives a score equal to or higher than the original, its quality is considered preserved.
- **JointOracle:** Similar to **SoloOracle**, but **Q** assigns numerical scores to both texts in the same prompt. The order is flipped in a second query. If the mutated text scores equal to or higher in both cases, quality is preserved.
- **RelativeOracle:** **Q** is prompted to select the better response or declare a tie, repeating the query with the order reversed. If the mutated text is chosen in both cases or a tie is declared, quality is preserved.
- **BinaryOracle:** **Q** is asked a direct yes/no question: “Is the mutated text just as good or better than the original?” If the response is “yes”, quality is preserved.
- **MutationOracle:** Similar to **BinaryOracle**, but the prompt explicitly states that the mutated text is a modification of the original. The query is repeated with the order reversed. If both responses are “yes”, quality is preserved.
- **ExampleOracle:** Similar to **BinaryOracle**, but includes an example (1-shot prompting) before presenting the actual texts. If the response is “yes”, quality is preserved.

- **DiffOracle:** Q is provided with the original text, mutated text, and a computed diff between them. It is asked whether these changes are acceptable. If the response is “yes”, quality is preserved.

These oracles serve as key components in our evaluation framework, allowing us to systematically assess how to best approximate human judgments of quality. By incorporating multiple prompting strategies, we ensure robustness in our analysis of watermark perturbation effectiveness.

8.7.11.2 Sandcastle Dataset Statistics

Since absolute quality scoring is difficult for humans [52], we formulated the annotation task as pairwise preference judgments with a tie option. Several coauthors, following standardized guidelines, compared perturbed texts to their originals, unaware of which was which. Table 8.10 shows the class distribution, where we merged "Attacked Better" and "Tie" into a Quality Preserved (QP) category to support binary classification.

Split	Quality Preserved		Quality Degraded		Total
	Attacked Better	Tie	Original Better		
Train	12	238	306		556
Test	1	103	135		239
Total	13	341	441		795

Table 8.10: Distribution of human quality assessments by split for the Sandcastles dataset. The table details counts for cases where attacked outputs were rated as "Attacked Better" or "Tie" (grouped under Quality Preserved (QP)) versus "Original Better", along with overall totals for both training and test sets.

8.7.12 Full Oracle Results

Table 8.11 provides a detailed comparison of quality oracles, including inference time, QP Precision, Overall F1, and RewardBench scores where available. Despite fine-tuning, no oracle fully aligns with human judgments, and high RewardBench scores do not guarantee strong performance in our setting. Proprietary models like GPT-4o with fine-tuning perform best but are impractical for large-scale attacks. Locally hosted models (MutationOracle, DiffOracle) offer a viable alternative but still misclassify degraded outputs. These results highlight the challenges of using LLM-based oracles for reliable watermark attack guidance.

Oracle	Model	Type	Time (s)	QP Prec.	Overall F1	RB Score
SkyworkOracle	Skywork-Reward-Gemma-2-27B-v0.2 [184]	FLOAT	2.22	43.51	26.39	94.3
RankOracle	Llama-3.1-70B-Instruct [78]	BOOL	4.33	50.00	37.09	–
SoloOracle	Llama-3.1-70B-Instruct [78]	INT	2.23	49.49	39.86	–
JointOracle	Llama-3.1-70B-Instruct [78]	INT	3.62	53.85	40.85	–
INFORMOracle	INF-ORM-Llama3.1-70B [212]	FLOAT	5.81	65.63	54.40	95.1
QRMOracle	QRM-Gemma-2-27B [77]	FLOAT	3.28	50.68	56.98	94.4
RelativeOracle	Llama-3.1-70B-Instruct [78]	CLASS	2.76	79.59	63.07	–
ExampleOracle	Llama-3.1-70B-Instruct [78]	BOOL	1.33	79.59	63.07	–
BinaryOracle	Llama-3.1-70B-Instruct [78]	BOOL	1.27	61.90	63.82	–
ArmoRMOOracle	ArmoRM-Llama3-8B-v0.1 [309]	FLOAT	0.33	65.71	64.26	90.4
OffsetBiasOracle	Llama-3-OffsetBias-RM-8B [236]	FLOAT	0.32	62.22	65.30	89.6
Prometheus2Absolute	prometheus-8x7b-v2.0 [143]	FLOAT	7.28	74.78	66.73	74.5
Prometheus2Relative	prometheus-8x7b-v2.0 [143]	BOOL	7.36	74.78	66.73	74.5
Prometheus2Absolute	GPT-4o [229]	INT	7.93	76.70	66.87	–
MutationOracle	Llama-3.1-70B-Instruct [78]	BOOL	2.74	84.62	66.93	–
Prometheus2Relative	GPT-4o [229]	BOOL	7.73	77.23	67.05	–
Prometheus2Relative	GPT-4-Turbo [228]	BOOL	11.94	75.00	67.27	–
Prometheus2Absolute	GPT-4-Turbo [228]	INT	12.46	76.15	67.55	–
InternLMOracle	internlm2-20b-reward [36]	FLOAT	0.86	65.69	69.84	90.6
DiffOracle	Llama-3.1-70B-Instruct [78]	BOOL	1.83	71.74	70.85	–
MutationOracle+FT	Llama-3.1-70B-Instruct + Fine-tuning	BOOL	3.25	81.18	71.83	–
DiffOracle+FT	Llama-3.1-70B-Instruct + Fine-tuning	BOOL	1.80	69.07	76.94	–
DiffOracle+FT	GPT-4o [229] + Fine-tuning	BOOL	0.46	75.51	77.32	–
MutationOracle+FT	GPT-4o [229] + Fine-tuning	BOOL	0.84	74.51	77.38	–

Table 8.11: Overview of oracle performance on our human-annotated test set. For each oracle we report average inference time, Quality-Preserved (QP) Precision, Overall F1, and RewardBench (RB) Score when available. Despite fine-tuning on human judgements, no oracle perfectly capture human quality assessments, and high RB Scores did not predict strong performance in our evaluation setting.

8.7.13 InternLM vs DiffOracle

We compared the proportion of cases where humans agreed that the quality of generated outputs was preserved. The results, summarized in Table 8.12, show that InternLMOracle had a higher agreement rate (47.78%) than DiffOracle+FT (40.0%).

Oracle	Agree QP	Disagree QP
DiffOracle	40.00	60.00
InternLM	47.78	52.22

Table 8.12: Comparison of human agreement rates on quality preservation (QP) percentages between DiffOracle and InternLM. InternLM shows a higher agreement rate, suggesting it aligns better with human judgments.

To quantify the probability that InternLMOracle is genuinely the better oracle, we adopt a Bayesian approach, modeling the probability of agreement for each oracle as a Beta distribution:

$$p_A \sim \text{Beta}(A + 1, N_A - A + 1),$$
$$p_B \sim \text{Beta}(B + 1, N_B - B + 1)$$

where A and B are the counts of human agreement for DiffOracle+FT and InternLMOracle, respectively, and N_A and N_B are the total evaluations for each oracle.

Using a Monte Carlo simulation with 100,000 samples, we estimate:

$$P(p_B > p_A) \approx 85.08\%$$

indicating that InternLMOracle has an 85.08% probability of being the better judge in preserving quality according to human evaluators. Given this high confidence, we justify the use of InternLMOracle as the preferred oracle for further evaluations⁴.

8.7.14 Human Annotation Details

Annotators were provided with the following instructions when reviewing:

⁴This surprising reversal of performance may be attributable to DiffOracle+FT managing too much noise in the changelog of edits when attacks exceed 20 steps (the maximum attack length present in the Sandcastles dataset).

- Determine which is a better response to the prompt: text *A*, text *B*, or tie.
- Judge quality based on content, style, cohesion, and prompt relevance.
- Note: Formatting is not especially important for quality (e.g. paragraph breaks should be ignored).

These guidelines ensured that evaluations focused on meaningful quality differences rather than superficial formatting artifacts.

8.7.15 Extended Attack Results Analysis

Table 8.13 provides a detailed breakdown of attack performance, including automated quality metrics, revealing several notable patterns. One interesting finding is that, in some cases, attacks appear to “improve” certain quality metrics, such as perplexity and grammar error rates. This effect is most pronounced for the Adaptive watermark, where the average perplexity and grammar errors decrease post-attack. However, this improvement is largely driven by a few low-quality outliers in the original watermarked dataset, rather than a systematic enhancement of text fluency. Despite these reductions in surface-level errors, the InternLM quality score consistently drops, indicating that attacks tend to reduce overall coherence and relevance, even when fluency-related metrics superficially improve.

Another trend is that unique bigram diversity (μ_{dt}) increases slightly in many cases, particularly for sentence- and document-level attacks. This suggests that perturbations introduce more varied word sequences, potentially disrupting structured patterns imposed by watermarking. However, this increase is relatively small, meaning that while attacks may inject lexical diversity, they do not necessarily enhance the text in a meaningful way. Instead, the most aggressive perturbation strategies—particularly sentence- and document-level attacks—cause the largest drops in the InternLM quality score, reinforcing the idea that these attacks are the most disruptive to text coherence. While they achieve the highest watermark removal rates, they also tend to introduce noticeable degradation, making the resulting text less natural and readable.

By contrast, perturbation strategies that fail to effectively break watermarks, such as word-level and entropy-based edits, also have minimal impact on quality metrics. This suggests that these finer-grained mutations are too minor to erase watermark signals while also being too weak to meaningfully degrade text fluency. More broadly, the average attack success rate remains relatively low even before enforcing quality constraints, with ASR_{fin} at only 26.13%. After accounting for quality degradation, this drops further to just 10.47%, confirming that successfully removing watermarks without compromising text quality remains a substantial challenge for adversaries.

Watermark	P Oracle	μ_{w_0}	μ_{w_t}	BP	ASR _{min}	ASR _{fin}	Reviewed	QP	$\neg$ QP	Q-ASR _{fin}	μ_{q_0}	μ_{q_t}	μ_{p_0}	μ_{p_t}	μ_{g_0}	μ_{g_t}	μ_{d_0}	μ_{d_t}
Adaptive	Word	99.27	70.37	56.16	0.00	0.00	0	0	0	0.00	0.45	0.01	63.32	77.64	16.21	13.79	603.64	609.09
Adaptive	EntropyWord	99.27	82.45	56.16	0.00	0.00	0	0	0	0.00	0.45	-0.05	63.32	78.01	16.21	19.00	603.64	608.33
Adaptive	Span	99.27	67.21	56.16	1.54	1.54	2	2	0	1.54	0.45	0.06	63.32	44.34	16.21	11.24	603.64	616.06
Adaptive	Sentence	99.27	59.93	56.16	35.34	19.21	20	8	12	7.68	0.45	0.24	63.32	27.78	16.21	5.42	603.64	602.26
Adaptive	Document	99.27	58.55	56.16	48.78	45.24	20	8	12	18.10	0.45	0.00	63.32	16.27	16.21	2.11	603.64	464.31
Adaptive	Document1Step	99.27	70.94	56.16	1.16	1.16	2	2	0	1.16	0.45	0.32	63.32	30.09	16.21	0.46	603.64	541.44
Adaptive	Document2Step	99.27	73.39	56.16	5.33	4.71	8	5	3	2.94	0.45	0.27	63.32	27.48	16.21	8.06	603.64	580.33
KGW	Word	0.28	0.17	0.21	47.54	20.00	20	4	16	4.00	0.16	-0.30	8.87	30.30	4.24	11.10	479.90	572.57
KGW	EntropyWord	0.28	0.22	0.21	3.45	0.56	1	0	1	0.00	0.16	-0.31	8.87	20.15	4.24	10.14	479.90	535.14
KGW	Span	0.28	0.20	0.21	38.46	32.35	20	14	6	22.65	0.16	-0.29	8.87	14.78	4.24	7.19	479.90	536.79
KGW	Sentence	0.28	0.14	0.21	89.47	56.52	20	7	13	19.78	0.16	0.02	8.87	16.93	4.24	4.68	479.90	658.71
KGW	Document	0.28	0.18	0.21	62.50	44.44	20	8	12	17.78	0.16	-0.27	8.87	10.48	4.24	7.70	479.90	435.39
KGW	Document1Step	0.28	0.27	0.21	12.66	8.54	14	7	7	4.27	0.16	0.07	8.87	9.47	4.24	0.31	479.90	482.51
KGW	Document2Step	0.28	0.18	0.21	9.09	7.78	10	4	6	3.11	0.16	-0.03	8.87	11.97	4.24	4.41	479.90	501.89
SIR	Word	5.32	1.74	1.27	78.22	57.89	20	1	19	2.89	0.43	0.00	9.56	29.96	2.86	6.79	512.44	588.33
SIR	EntropyWord	5.32	3.30	1.27	39.68	27.54	20	0	20	0.00	0.43	-0.03	9.56	20.20	2.86	1.44	512.44	554.38
SIR	Span	5.32	1.57	1.27	60.71	37.40	20	5	15	9.35	0.43	0.03	9.56	15.28	2.86	4.36	512.44	561.04
SIR	Sentence	5.32	0.52	1.27	87.65	74.71	20	13	7	48.56	0.43	0.30	9.56	17.97	2.86	2.93	512.44	597.44
SIR	Document	5.32	0.93	1.27	61.54	46.09	20	6	14	13.83	0.43	0.10	9.56	12.65	2.86	10.64	512.44	454.21
SIR	Document1Step	5.32	2.54	1.27	14.04	14.04	12	11	1	12.87	0.43	0.23	9.56	11.74	2.86	0.26	512.44	491.45
SIR	Document2Step	5.32	3.07	1.27	68.09	49.06	20	12	8	29.44	0.43	0.26	9.56	11.70	2.86	4.29	512.44	530.79
Averages					36.44	26.13		40.48	59.52	10.47	0.35	0.03	27.25	25.49	7.77	6.49	532.00	548.69

Table 8.13: Attack success rates (ASR) and automated quality scores across different perturbation strategies. Human review reveals an average of 59.52% of successfully attacked texts have degraded quality. μ_{w_0} represents the initial watermark score at step 0, while μ_{w_t} represents the final watermark score after t mutation steps. “min” refers to the point where the watermark score is at its lowest during the attack while “fin” refers to score at the final step of the attack. “Reviewed” indicates the number of human-reviewed examples where the watermark was broken. **QP** and $\neg$ **QP** represent the number of cases where human reviewers judged the attacked text as quality-preserving or degraded, respectively. **Q-ASR_{fin}** is the re-estimated attack success after controlling for quality, calculated as $\text{ASR}_{\text{fin}} \times (\text{QP}/\text{Reviewed})$. The remaining quality columns show, for time step 0 and the final step t , InterLM quality score (q), perplexity (p), grammar errors (g), and unique bigram diversity (d). On average, quality degraded significantly while perplexity, grammar errors, and diversity improved.

8.7.16 Attack Success Rate vs. Detection Threshold

Figures 8.7, 8.6, and 8.5 plot the final attack success rate (**ASR**) for each perturbation oracle under the **KGW**, **SIR**, and **Adaptive** watermarking schemes, respectively. The horizontal axis represents detection thresholds measured in standard deviations above the mean detection score for unwatermarked text (i.e., 0σ , 1σ , 2σ , or 3σ). A higher threshold allows more texts to be considered “unwatermarked,” so ASR generally increases as we move to the right. The vertical axis indicates the fraction of attacked texts that fall below each threshold once all permitted mutations have been applied.

Each curve corresponds to a specific mutator—Word, EntropyWord, Span, Sentence, Document, Document1Step, or Document2Step—with line style distinguishing s_{min} (dotted) from s_{fin} (solid). In general, token-level **P** (Word, EntropyWord, Span) make smaller, more localized edits, while document-level **P** (Document, Document1Step, Document2Step) can restructure larger portions of text. Comparing these curves reveals which **P** achieve higher ASR for each watermarking scheme and how sensitive those results are to stricter or looser detection thresholds.

Overall, two main patterns emerge. First, as the detection threshold increases, more perturbed texts evade being flagged, causing the ASR curves to rise. Second, the extent of this rise varies across both watermarking schemes and **P**: some methods prove more effective at evading detection for KGW or SIR, whereas Adaptive typically shows lower ASR across thresholds. This aligns with our broader observation that larger, more context-aware edits (Document-based **P**) often outperform smaller, token-level edits, but still rarely achieve high success rates without risking noticeable quality degradation.

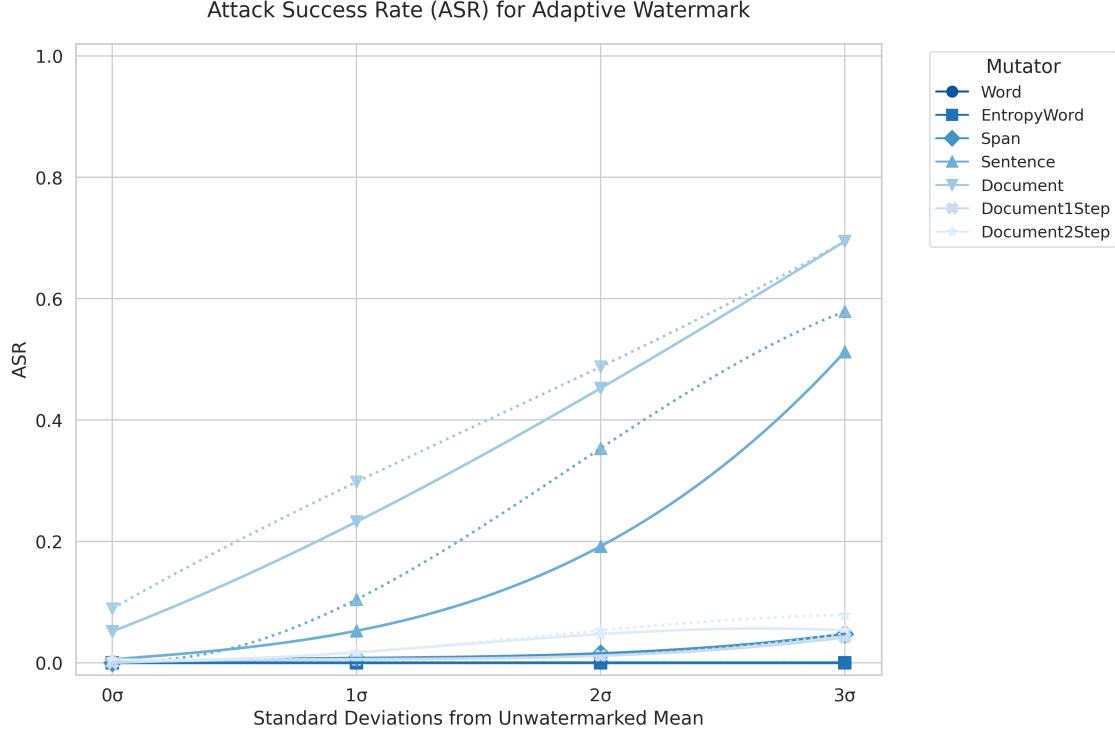


Figure 8.5: Attack success rate (ASR) vs. detection threshold for the Adaptive watermarking scheme. Each curve represents a different perturbation oracle, with thresholds measured in standard deviations above the unwatermarked mean.

8.7.17 What factors contributed to attack inefficiency?

The efficiency of the WITS attack against private watermarking schemes is hampered by two interrelated challenges. First, the attack relies on a random walk that must approach its stationary distribution, with the mixing time critically dependent on the second-largest eigenvalue, g , of the transition matrix $\vec{P}$. Not only is computing g exactly infeasible, but even approximating it is extremely difficult. In practice, the size and complexity of $\vec{P}$ which depends on factors such as the mutator, prompt, and quality barrier make computing any information about $\vec{P}$ computationally intractable. As a result, the attacker must rely on upper bounds for g to estimate the mixing time, a strategy that introduces significant uncertainty into the overall attack duration. Notice that this isn't an issue for public watermarking schemes since the attacker can stop as soon as the watermark is removed.

Second, attempts to accelerate the mixing process such as by increasing the step size of the perturbation oracle risk degrading the quality of the text. As quality decreases, so does the success

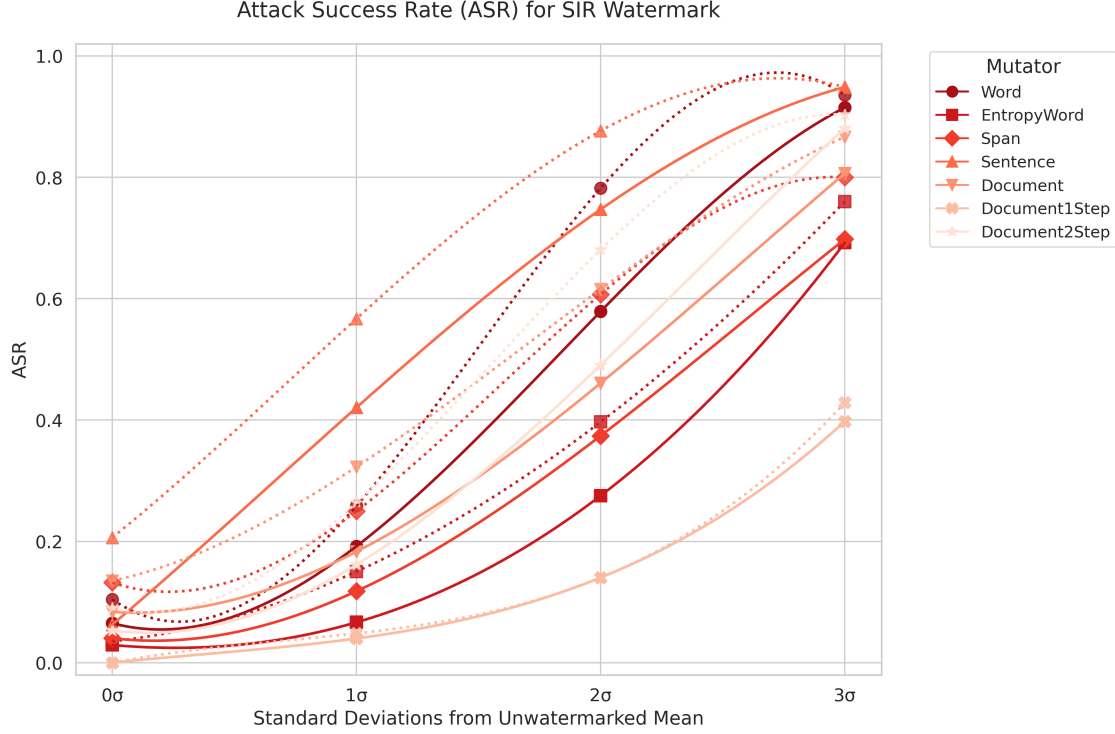


Figure 8.6: Attack success rate (ASR) vs. detection threshold for the SIR watermarking scheme. The plot shows the fraction of attacked texts falling below various thresholds (in standard deviations above the unwatermarked mean) for multiple perturbation oracles.

rate of mutations (i.e., the effective constant ϵ_{pert} no longer holds), which in turn negates the benefits of improved mixing by requiring even more iterations to produce acceptable outputs.

In essence, there is a fundamental tension between reducing the mixing time to achieve attack efficiency and maintaining the quality of the attacked text. A more refined theoretical analysis that balances these competing factors is necessary to fully understand the capabilities of the WITS attack. We leave this compelling direction for future work.

Figures 8.8, 8.9, 8.10, and 8.11 below illustrate the rolling success rate of mutations across various watermarking schemes and mutator types, thereby supporting our first claim. In these computations, the window size is defined as one-tenth of the total number of mutator steps (e.g., for the Sentence Mutator, $150/10 = 15$ steps).

Notably, **P** characterized by larger step sizes exhibit lower success rates. Furthermore, the plots reveal a modest correlation between the mutation success rate and the entropy level: prompts with lower entropy tend to have reduced success rates. This phenomenon may be attributable to the

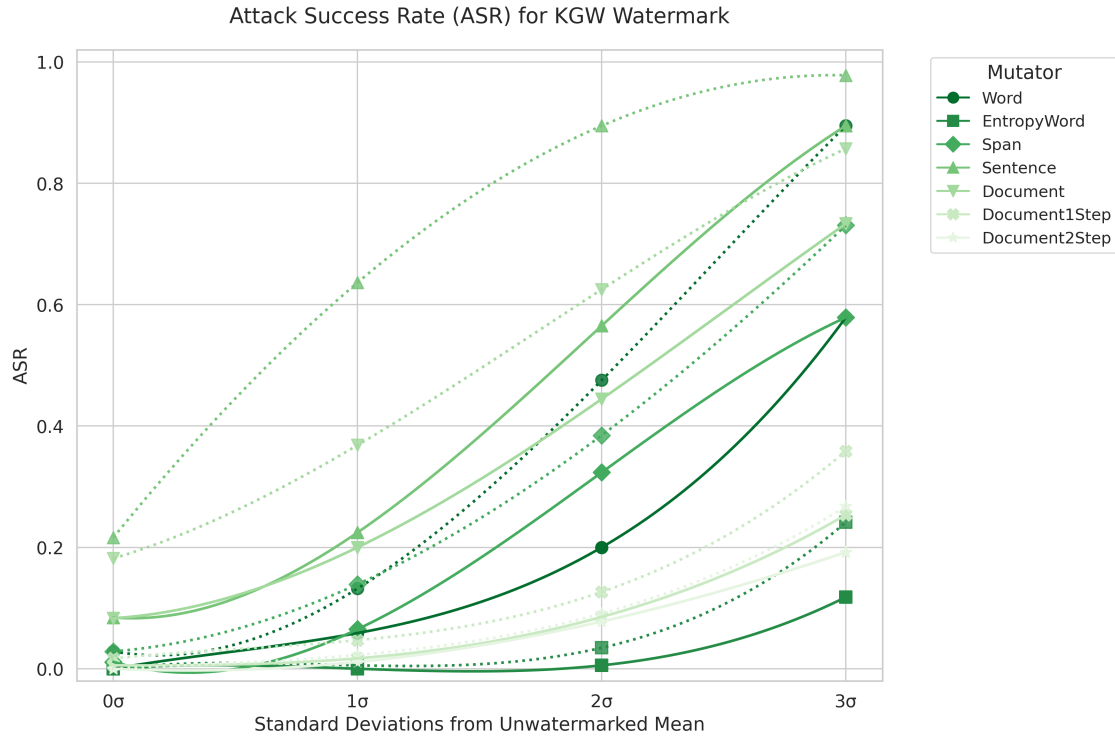


Figure 8.7: Attack success rate (ASR) vs. detection threshold for the KGW watermarking scheme. Different curves correspond to various perturbation oracles, with the detection threshold defined as standard deviations above the mean detection score of unwatermarked texts.

fact that lower-entropy prompts are generally longer, thereby increasing the difficulty of generating a mutated response that maintains high quality. Consequently, any interpretation of this correlation should be approached with caution.

Rolling Success Rate of Mutations Over Time (GPT4o_unwatermarked)

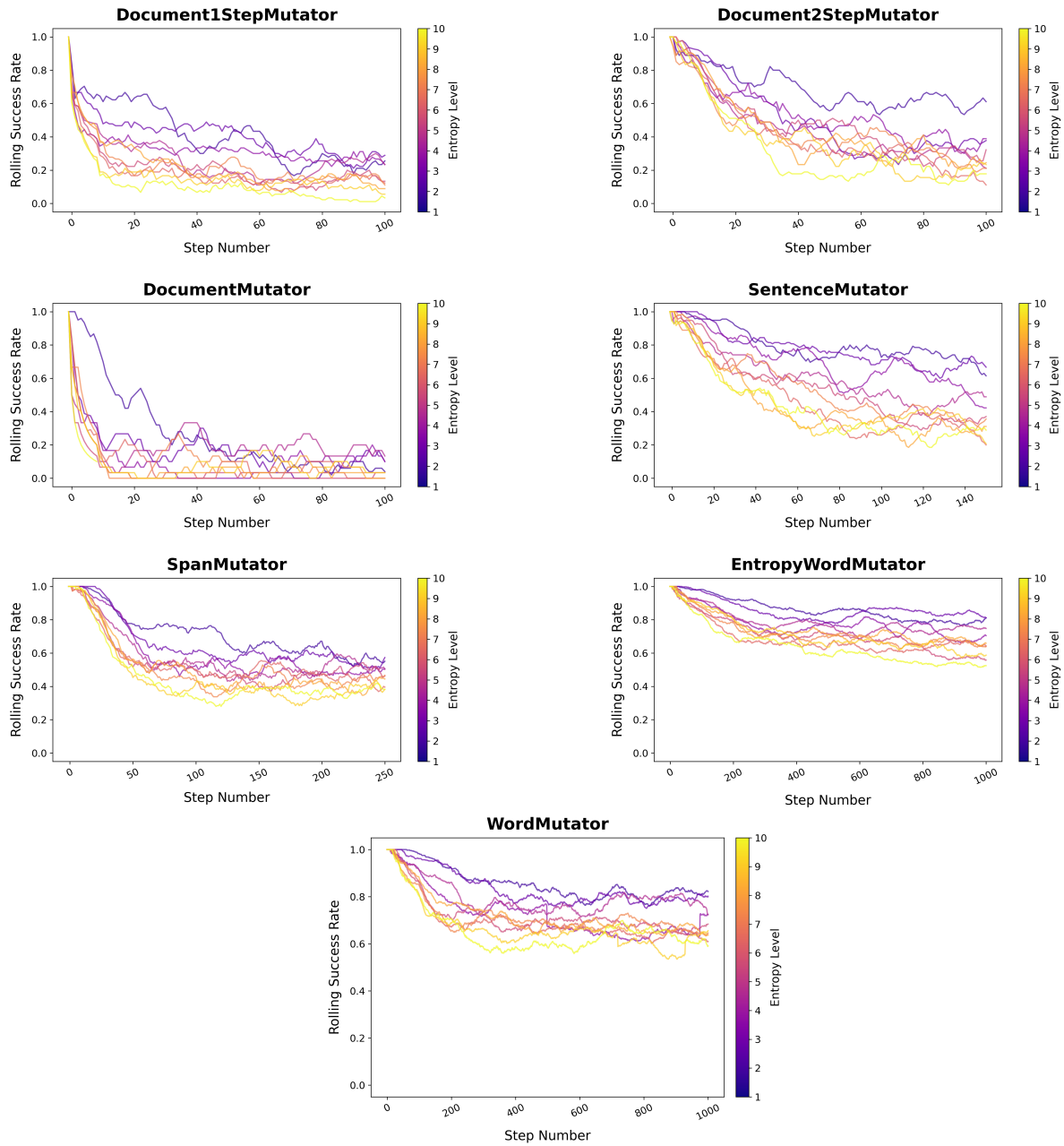


Figure 8.8: Rolling success rate for GPT-4o generations, which are unwatermarked.

Rolling Success Rate of Mutations Over Time (KGW)

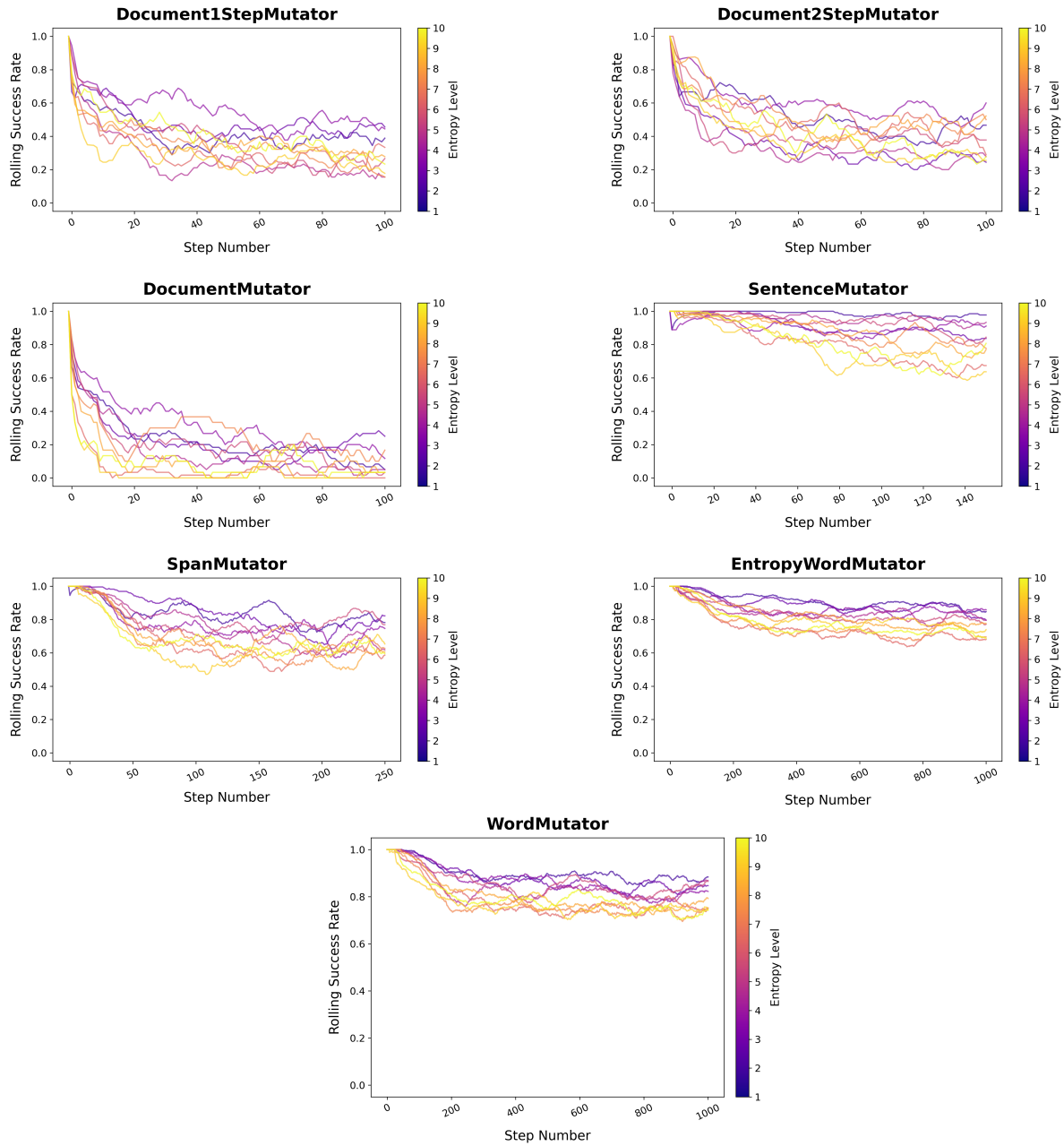


Figure 8.9: Rolling success rate for the KGW watermark.

Rolling Success Rate of Mutations Over Time (SIR)

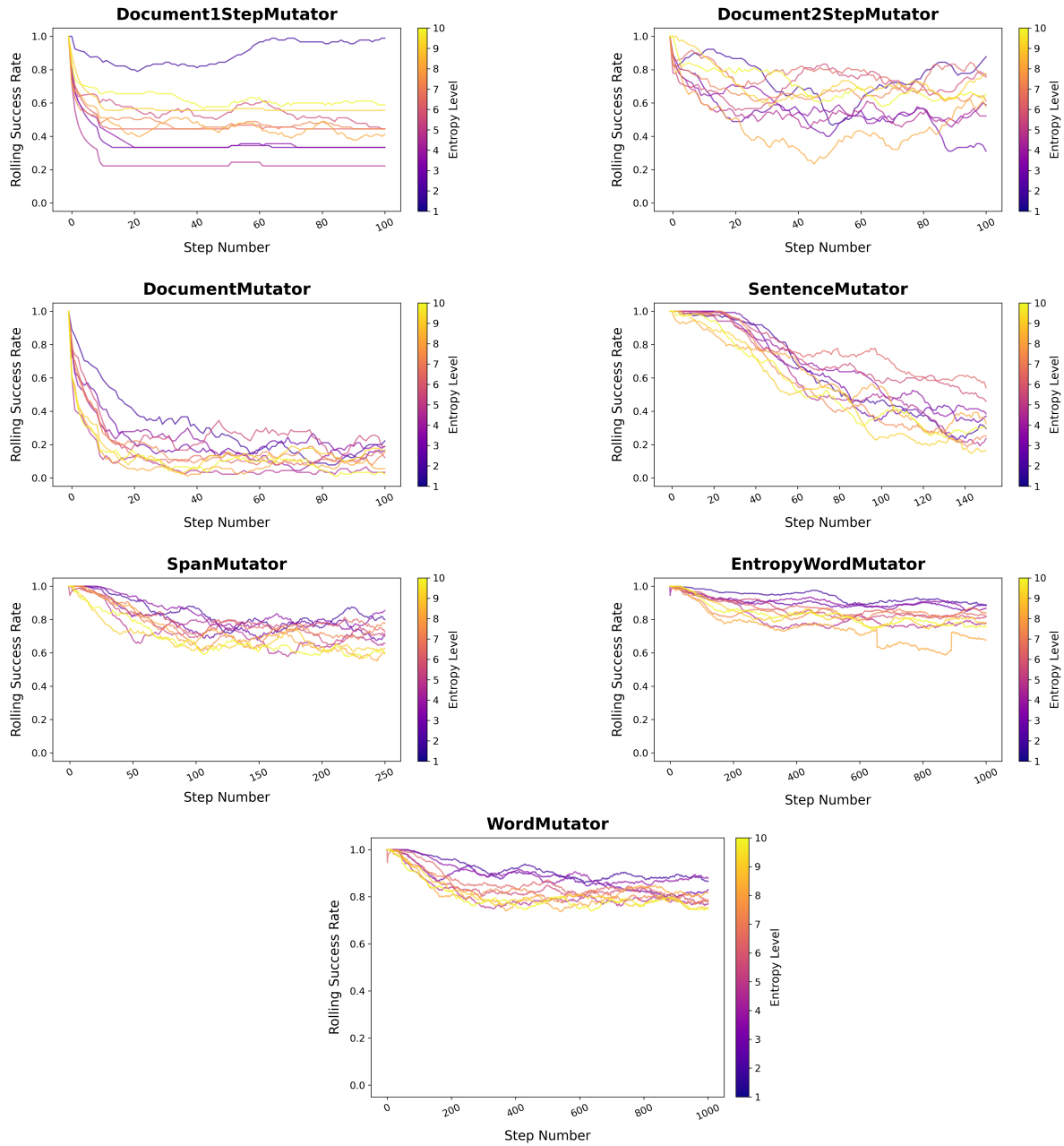


Figure 8.10: Rolling success rate for the SIR watermark.

Rolling Success Rate of Mutations Over Time (Adaptive)

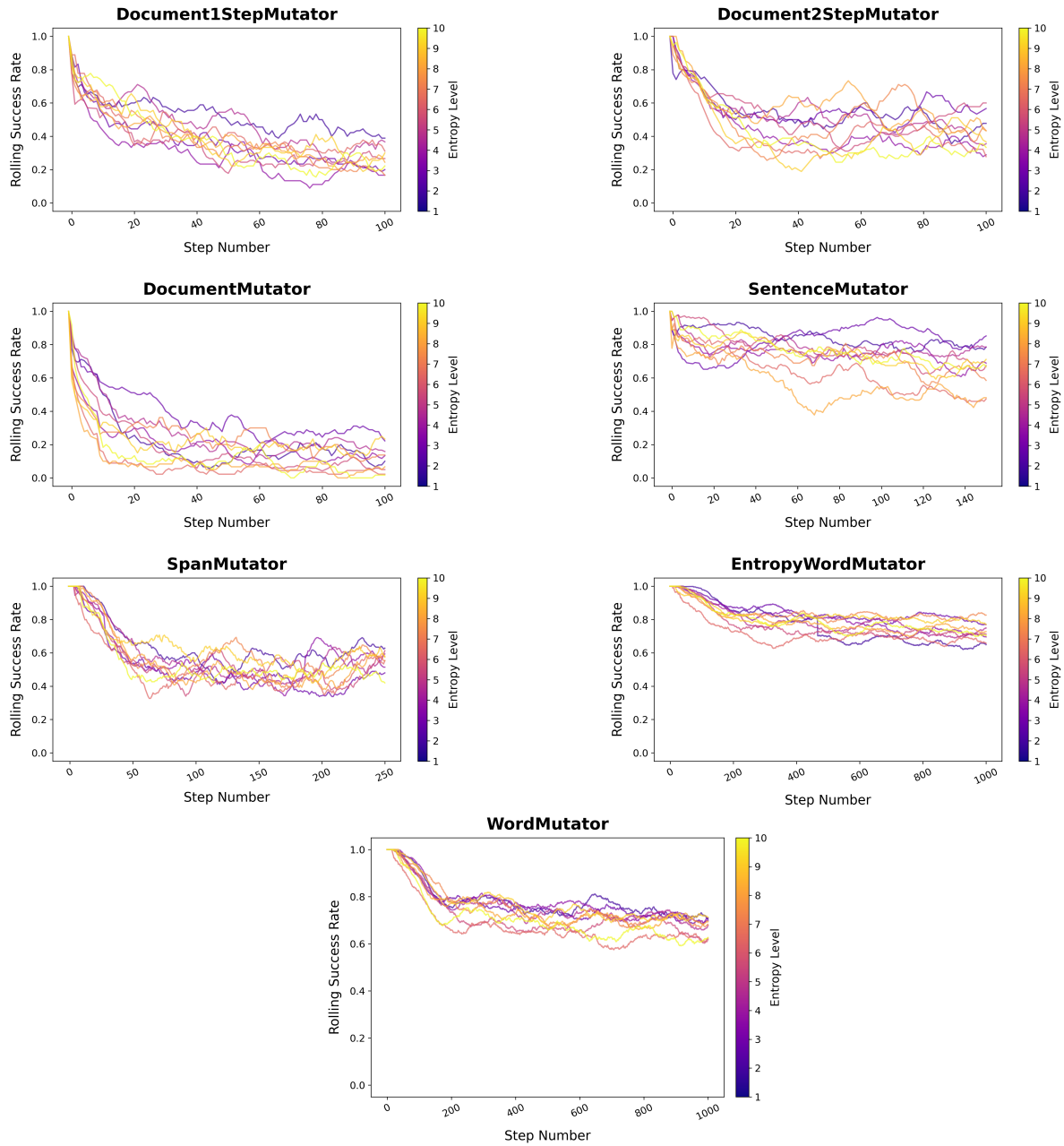


Figure 8.11: Rolling success rate for the Adaptive watermark.

CHAPTER 9

Conclusion

This dissertation began by confronting a critical challenge in the rapidly evolving field of artificial intelligence: evaluation frameworks have struggled to keep pace with the increasing sophistication of ML systems, particularly language models. Traditional metrics, often narrowly focused on structural or quantitative aspects, fail to capture the multifaceted nature of AI performance, neglecting crucial dimensions like data diversity, subjective quality, and provenance. The central thesis advanced throughout this work is that robust, meaningful AI evaluation demands a paradigm shift away from these limited approaches towards holistic frameworks that harmoniously integrate data diversity, human-centric subjectivity, and reliable methods for authorship verification. By embracing these pillars, we can develop a more comprehensive and insightful understanding of AI capabilities and limitations, ultimately guiding the development of more reliable, useful, and trustworthy systems.

The research presented herein systematically explored these pillars through distinct but interconnected investigations. Initial studies cast significant doubt on the efficacy of purely structural metrics like neuron coverage when applied to deep neural networks (RQ1), demonstrating their weak correlation with practical defect detection and diverse test suite generation. This underscored the need for data-centric perspectives, leading to the development of sibylvariant transformations (Sibyl), which powerfully enhanced data diversity and model robustness by strategically relaxing label-preserving constraints (RQ2). To manage the complexities of augmentation, the FADA framework and the INSPECTOR tool provided methods for optimizing augmentation policies using semantic features and enabling efficient, provenance-aware human validation, ensuring both diversity and quality (RQ3). Recognizing that technical correctness is insufficient, the Psychological Depth Scale (PDS) introduced a novel methodology for quantifying the subjective, experiential

impact of AI-generated text on human readers (RQ4), revealing AI's striking ability to mimic human-like qualities. Finally, addressing the implications of such high-fidelity generation, the analysis of watermarking techniques affirmed their practical resilience and necessity for verifying authorship in an era where distinguishing human from machine output is increasingly challenging (RQ5).

Collectively, these studies demonstrate that a multi-faceted approach is indispensable for evaluating modern AI. Relying solely on structural coverage is insufficient; we must actively cultivate and measure data diversity. Optimizing models requires not just more data, but better data, managed through intelligent augmentation and careful inspection informed by provenance and human judgment. Furthermore, particularly for generative models, evaluation must encompass the subjective human experience, measuring qualities like empathy and engagement that define real-world impact. Finally, as AI capabilities converge with human abilities, mechanisms for verifying authorship become non-negotiable for maintaining trust and accountability. This dissertation provides evidence and methodologies supporting each of these dimensions, contributing components to a more unified and human-centric evaluation paradigm.

Ultimately, the transition towards this holistic evaluation framework is essential for guiding the responsible development and deployment of artificial intelligence. By moving beyond simplistic metrics and embracing the complexities of data diversity, subjective experience, and provenance, we equip ourselves with the tools needed to assess AI systems not just for their technical prowess, but for their alignment with human values, their reliability in real-world contexts, and their overall contribution to society. The insights and methods presented here offer a foundation for this shift, paving the way for future work in refining these techniques and fostering an AI ecosystem that is both powerful and trustworthy.

Bibliography

- [1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Al-tenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report, 2023.
- [2] N. Alshahwan and M. Harman. Coverage and fault detection of the output-uniqueness test selection criteria. In *Proceedings of the 2014 International Symposium on Software Testing and Analysis*, ISSTA 2014, page 181192, New York, NY, USA, 2014. Association for Computing Machinery.
- [3] M. Alzantot, Y. Sharma, A. Elgohary, B. Ho, M. B. Srivastava, and K. Chang. Generating natural language adversarial examples. *CoRR*, abs/1804.07998, 2018.
- [4] T. M. Amabile. Social psychology of creativity: A consensual assessment technique. *Journal of personality and social psychology*, 43(5):997, 1982.
- [5] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling. Not enough data? deep learning to the rescue!, 2019.
- [6] Anthropic. Introducing claude, Mar 2023.
- [7] Z. Ashktorab, M. Desmond, J. Andres, M. Muller, N. N. Joshi, M. Brachman, A. Sharma, K. Brimijoin, Q. Pan, C. T. Wolf, et al. AI-assisted human labeling: Batching for efficiency without overreliance. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–27, 2021.
- [8] R. Babaei and I. Montashery. Stanley fish with respect to the reader. *International Journal of English and Literature*, 3:18–24, 2012.
- [9] J. Baer. Why you are probably more creative (and less creative) than you think. In *The creative self*, pages 259–273. Elsevier, 2017.
- [10] J. Baer and J. C. Kaufman. Assessing creativity with the consensual assessment technique. *The Palgrave handbook of social creativity research*, pages 27–37, 2019.

- [11] J. Baer and S. S. McKool. The gold standard for assessing creativity. *International Journal of Quality Assurance in Engineering and Technology Education (IJQAETE)*, 3(1):81–93, 2014.
- [12] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.
- [13] P. M. Bal and M. Veltkamp. How does fiction reading influence empathy? an experimental investigation on the role of emotional transportation. *PLOS ONE*, 8(1):1–12, 01 2013.
- [14] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. Abstract meaning representation (AMR) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*, pages 1533–1544, 2012.
- [15] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. Abstract Meaning Representation for semantic banking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics.
- [16] G. Bansal, T. Wu, J. Zhou, R. Fok, B. Nushi, E. Kamar, M. T. Ribeiro, and D. Weld. Does the whole exceed its parts? the effect of AI explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.
- [17] F. Barbieri, J. Camacho-Collados, L. E. Anke, and L. Neves. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, 2020.
- [18] E. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo. The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering*, 41:507–525, 2015.

- [19] S. Barratt. *Inception Score for GANs in Pytorch*, 2018 (accessed August 8, 2019).
- [20] L. Barrett, M. Lewis, and J. Haviland-Jones. *Handbook of Emotions, Fourth Edition*. Psychology (The Guilford Press). Guilford Publications, 2016.
- [21] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, page 610623, New York, NY, USA, 2021. Association for Computing Machinery.
- [22] I. Berlin. *Three Critics of the Enlightenment: Vico, Hamann, Herder*. Princeton University Press, Princeton, 2000.
- [23] P. Bhargava, A. Drozd, and A. Rogers. Generalization in NLI: Ways (not) to go beyond simple heuristics. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.
- [24] bjascob. amrlib, 2023.
- [25] S. Black, G. Leo, P. Wang, C. Leahy, and S. Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, Mar. 2021. If you use this software, please cite it using these metadata.
- [26] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.
- [27] K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.
- [28] B. Brosgol. Do-178c: The next avionics safety standard. In *Proceedings of the 2011 ACM Annual International Conference on Special Interest Group on the Ada Programming Language, SIGAda '11*, pages 5–6, New York, NY, USA, 2011. ACM.

- [29] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krüger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.
- [30] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [31] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.
- [32] S. M. Bsharat, A. Myrzakhan, and Z. Shen. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4, 2024.
- [33] R. Busselle and H. Bilandzic. Fictionality and perceived realism in experiencing stories: A model of narrative comprehension and engagement. *Communication theory*, 18(2):255–280, 2008.
- [34] R. Busselle and H. Bilandzic. Measuring narrative engagement. *Media psychology*, 12(4):321–347, 2009.
- [35] Á. A. Cabrera, E. Fu, D. Bertucci, K. Holstein, A. Talwalkar, J. I. Hong, and A. Perer. Zeno: An interactive framework for behavioral evaluation of machine learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2023.
- [36] Z. Cai, M. Cao, H. Chen, K. Chen, K. Chen, X. Chen, X. Chen, Z. Chen, Z. Chen, P. Chu, X. Dong, H. Duan, Q. Fan, Z. Fei, Y. Gao, J. Ge, C. Gu, Y. Gu, T. Gui, A. Guo, Q. Guo,

- C. He, Y. Hu, T. Huang, T. Jiang, P. Jiao, Z. Jin, Z. Lei, J. Li, J. Li, L. Li, S. Li, W. Li, Y. Li, H. Liu, J. Liu, J. Hong, K. Liu, K. Liu, X. Liu, C. Lv, H. Lv, K. Lv, L. Ma, R. Ma, Z. Ma, W. Ning, L. Ouyang, J. Qiu, Y. Qu, F. Shang, Y. Shao, D. Song, Z. Song, Z. Sui, P. Sun, Y. Sun, H. Tang, B. Wang, G. Wang, J. Wang, J. Wang, R. Wang, Y. Wang, Z. Wang, X. Wei, Q. Weng, F. Wu, Y. Xiong, C. Xu, R. Xu, H. Yan, Y. Yan, X. Yang, H. Ye, H. Ying, J. Yu, J. Yu, Y. Zang, C. Zhang, L. Zhang, P. Zhang, P. Zhang, R. Zhang, S. Zhang, S. Zhang, W. Zhang, W. Zhang, X. Zhang, X. Zhang, H. Zhao, Q. Zhao, X. Zhao, F. Zhou, Z. Zhou, J. Zhuo, Y. Zou, X. Qiu, Y. Qiao, and D. Lin. Internlm2 technical report, 2024.
- [37] M. Cakmak, C. Chao, and A. L. Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010.
- [38] P. Canning. Text world theory and real world readers: From literature to life in a belfast prison. *Language and Literature*, 26(2):172–187, 2017. PMID: 29278261.
- [39] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2016.
- [40] S. H. Carson. 14 creativity and mental illness. *Cambridge Handbook of Creativity*, pages 296–318, 2019.
- [41] T. Chakrabarty, P. Laban, D. Agarwal, S. Muresan, and C.-S. Wu. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI ’24, New York, NY, USA, 2024. Association for Computing Machinery.
- [42] L. Chan. Reader response and reception theory, 2016.
- [43] J. C. Chang, S. Amershi, and E. Kamar. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2334–2346, 2017.
- [44] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. In T. Leen,

- T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.
- [45] H. Chase. LangChain, Oct. 2022.
- [46] C. Chen, X. Gong, Z. Liu, W. Jiang, S. Q. Goh, and K.-Y. Lam. Trustworthy, responsible, and safe ai: A comprehensive architectural framework for ai safety with challenges and mitigations. *arXiv preprint arXiv:2408.12935*, 2024.
- [47] C. Chen and K. Shu. Can LLM-generated misinformation be detected? In *The Twelfth International Conference on Learning Representations*, 2024.
- [48] J. Chen, Z. Yang, and D. Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *CoRR*, abs/2004.12239, 2020.
- [49] J. Chen, Z. Yang, and D. Yang. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online, July 2020. Association for Computational Linguistics.
- [50] T. Chen, S. Cheung, and S. Yiu. Metamorphic testing: A new approach for generating next test cases. *ArXiv*, abs/2002.12543, 2020.
- [51] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.
- [52] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, H. Zhang, B. Zhu, M. Jordan, J. E. Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.
- [53] D. Choi, S. Evensen, Ç. Demiralp, and E. Hruschka. Tagruler: Interactive tool for span-level data programming by demonstration. In *Companion Proceedings of the Web Conference 2021*, pages 673–677, 2021.

- [54] M. Choi, C. Park, S. Yang, Y. Kim, J. Choo, and S. R. Hong. AILA: Attentive interactive labeling assistant for document classification through attention-based deep neural networks. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12, 2019.
- [55] D. Chopard, M. S. Treder, and I. Spasić. Learning data augmentation schedules for natural language processing. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 89–102, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.
- [56] A. Chouldechova and A. Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.
- [57] Y. Chung, T. Kraska, N. Polyzotis, K. H. Tae, and S. E. Whang. Automated data slicing for model validation: A big data-AI integration approach. *IEEE Transactions on Knowledge and Data Engineering*, 32(12):2284–2296, 2019.
- [58] F. Citron, M. Gray, H. Critchley, B. Weekes, and E. Ferstl. Emotional valence and arousal affect reading in an interactive way: Neuroimaging evidence for an approach-withdrawal framework. *Neuropsychologia*, 56:79–89, 04 2014.
- [59] E. Clark-Polner, T. D. Wager, A. B. Satpute, and L. F. Barrett. Neural fingerprinting: Meta-analysis, variation and the search for brain-based essences in the science of emotion. *The handbook of emotion*, pages 146–65, 2016.
- [60] cleanlab.ai. cleanlab, 2023.
- [61] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [62] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 113–123. Computer Vision Foundation / IEEE, 2019.

- [63] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719, 2019.
- [64] J. Cutting. Simplicity, complexity, and narration in popular movies, 08 2019.
- [65] M. Davis. *Empathy: A Social Psychological Approach*. Brown & Benchmark’s social psychology series. Brown & Benchmark Publishers, 1994.
- [66] M. H. Davis et al. A multidimensional approach to individual differences in empathy, 1980.
- [67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [68] M. Desmond, M. Muller, Z. Ashktorab, C. Dugan, E. Duesterwald, K. Brimijoin, C. Finegan-Dollak, M. Brachman, A. Sharma, N. N. Joshi, et al. Increasing the speed and accuracy of data labeling through an AI assisted interface. In *26th International Conference on Intelligent User Interfaces*, pages 392–401, 2021.
- [69] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [70] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [71] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, June 2019. Association for Computational Linguistics.
- [72] T. DeVries and G. W. Taylor. Dataset augmentation in feature space, 2017.
- [73] T. Devries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017.

- [74] K. D. Dhole, V. Gangal, S. Gehrmann, A. Gupta, Z. Li, S. Mahamood, A. Mahendiran, S. Mille, A. Shrivastava, S. Tan, T. Wu, J. Sohl-Dickstein, J. D. Choi, E. Hovy, O. Dusek, S. Ruder, S. Anand, N. Aneja, R. Banjade, L. Barthe, H. Behnke, I. Berlot-Attwell, C. Boyle, C. Brun, M. A. S. Cabezudo, S. Cahyawijaya, E. Chapuis, W. Che, M. Choudhary, C. Clauss, P. Colombo, F. Cornell, G. Dagan, M. Das, T. Dixit, T. Dopierre, P.-A. Dray, S. Dubey, T. Ekeinhor, M. D. Giovanni, T. Goyal, R. Gupta, R. Gupta, L. Hamla, S. Han, F. Harel-Canada, A. Honore, I. Jindal, P. K. Joniak, D. Kleyko, V. Kovatchev, K. Krishna, A. Kumar, S. Langer, S. R. Lee, C. J. Levinson, H. Liang, K. Liang, Z. Liu, A. Lukyanenko, V. Marivate, G. de Melo, S. Meoni, M. Meyer, A. Mir, N. S. Moosavi, N. Muennighoff, T. S. H. Mun, K. Murray, M. Namysl, M. Obedkova, P. Oli, N. Pasricha, J. Pfister, R. Plant, V. Prabhu, V. Pais, L. Qin, S. Raji, P. K. Rajpoot, V. Raunak, R. Rinberg, N. Roberts, J. D. Rodriguez, C. Roux, V. P. H. S., A. B. Sai, R. M. Schmidt, T. Scialom, T. Sefara, S. N. Shamsi, X. Shen, H. Shi, Y. Shi, A. Shvets, N. Siegel, D. Sileo, J. Simon, C. Singh, R. Sitelew, P. Soni, T. Sorensen, W. Soto, A. Srivastava, K. A. Srivatsa, T. Sun, M. V. T, A. Tabassum, F. A. Tan, R. Teehan, M. Tiwari, M. Tolkiehn, A. Wang, Z. Wang, G. Wang, Z. J. Wang, F. Wei, B. Wilie, G. I. Winata, X. Wu, W. Wydmaski, T. Xie, U. Yaseen, M. A. Yee, J. Zhang, and Y. Zhang. Nl-augmenter: A framework for task-sensitive natural language augmentation, 2022.
- [75] J. Dodge, M. Sap, A. Marasovic, W. Agnew, G. Ilharco, D. Groeneveld, and M. Gardner. Documenting the english colossal clean crawled corpus. *CoRR*, abs/2104.08758, 2021.
- [76] Y. Dong, P. Zhang, J. Wang, S. Liu, J. Sun, J. Hao, X. Wang, L. Wang, J. S. Dong, and D. Ting. There is limited correlation between coverage and robustness for deep neural networks. *arXiv preprint arXiv:1911.05904*, 2019.
- [77] N. Dorka. Quantile regression for distributional reward models in rlhf. *arXiv preprint arXiv:2409.10164*, 2024.
- [78] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

- [79] J. J. Dudley and P. O. Kristensson. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):1–37, 2018.
- [80] S. Edunov, M. Ott, M. Auli, and D. Grangier. Understanding back-translation at scale, 2018.
- [81] W. Empson. *The Structure of Complex Words*. Chatto and Windus, London, 1947.
- [82] S. Evensen, C. Ge, D. Choi, and Ç. Demiralp. Data programming by demonstration: A framework for interactively learning labeling functions. *arXiv preprint arXiv:2009.01444*, 2020.
- [83] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. A survey of data augmentation approaches for nlp, 2021.
- [84] S. Fish. Literature in the reader: Affective stylistics. *New Literary History*, 2(1):123–162, 1970.
- [85] M. N. Forster. Johann gottfried von herder. <https://plato.stanford.edu/entries/herder>, 2022. Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.).
- [86] S. Galhotra, Y. Brun, and A. Meliou. Fairness testing: Testing software for discrimination, 08 2017.
- [87] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. *CoRR*, abs/1801.04354, 2018.
- [88] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2018.
- [89] J. Gavins. *Text World Theory: An Introduction*. Edinburgh University Press, 2007.
- [90] J. Gavins. *Text world theory: An introduction*. Edinburgh University Press, 2007.
- [91] S. Gehrmann, T. Adewumi, K. Aggarwal, P. S. Ammanamanchi, A. Anuoluwapo, A. Bosse-lut, K. R. Chandu, M. Clinciu, D. Das, K. D. Dhole, et al. The GEM benchmark: Natural

- language generation, its evaluation and metrics. In A. Bosselut, E. Durmus, V. P. Gangal, S. Gehrmann, Y. Jernite, L. Perez-Beltrachini, S. Shaikh, and W. Xu, editors, *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online, Aug. 2021. Association for Computational Linguistics.
- [92] K. E. Gerdes, E. A. Segal, and C. A. Lietz. Conceptualising and measuring empathy. *British Journal of Social Work*, 40(7):2326–2343, 2010.
- [93] R. Gerrig. *Experiencing narrative worlds*. Routledge, 1993.
- [94] S. Ghazarian, R. Weischedel, A. Galstyan, and N. Peng. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems, 2020.
- [95] F. Gilardi, M. Alizadeh, and M. Kubli. ChatGPT outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*, 2023.
- [96] E. Goffman. Fun in games. In *Encounters: Two Studies in the Sociology of Interaction*, pages 15–81. Bobbs-Merrill, Indianapolis, 1961.
- [97] S. Goldfarb-Tarrant, H. Feng, and N. Peng. Plan, write, and revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 89–97, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [98] C. Gómez-Rodríguez and P. Williams. A confederacy of models: a comprehensive evaluation of LLMs on creative writing. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14504–14528, Singapore, Dec. 2023. Association for Computational Linguistics.
- [99] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS’14*, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

- [100] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014.
- [101] A. M. Grant. Does intrinsic motivation fuel the prosocial fire? motivational synergy in predicting persistence, performance, and productivity. *Journal of applied psychology*, 93(1):48, 2008.
- [102] M. Green and T. Brock. The role of transportation in the persuasiveness of public narrative. *Journal of personality and social psychology*, 79:701–21, 11 2000.
- [103] M. Green, S. Kass, J. Carrey, R. Feeney, B. Herzig, and J. Sabini. Transportation across media: Print versus film comparisons. *Media Psychology*, 11(4):512–539, 2008.
- [104] M. C. Green. Transportation into narrative worlds: The role of prior knowledge and perceived realism. *Discourse Processes*, 38(2):247–266, 2004.
- [105] H. Guo. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *AAAI*, 2020.
- [106] H. Guo, Y. Mao, and R. Zhang. Augmenting data with mixup for sentence classification: An empirical study. *CoRR*, abs/1905.08941, 2019.
- [107] A. Hall. Reading realism: Audiences’ evaluations of the reality of media texts. *Journal of Communication*, 53:624 – 641, 12 2003.
- [108] D. L. Hansen, P. J. Schone, D. Corey, M. Reid, and J. Gehring. Quality control mechanisms for crowdsourcing: peer review, arbitration, & expertise at familysearch indexing. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 649–660, 2013.
- [109] F. Harel-Canada, M. A. Gulzar, N. Peng, and M. Kim. Sibylvariant transformations for robust text classification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1771–1788, Dublin, Ireland, May 2022. Association for Computational Linguistics.

- [110] F. Harel-Canada, M. A. Gulzar, N. Peng, and M. Kim. Sibylvariant transformations for robust text classification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1771–1788, 2022.
- [111] K. J. Hayhurst. *A practical tutorial on modified condition/decision coverage*. DIANE Publishing, 2001.
- [112] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.
- [113] M. Heidegger. *Being and Time*. Harper & Row, New York, 1962. Originally published in 1927. Translated by J. Macquarrie & E. Robinson.
- [114] D. Hendrycks and K. Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016.
- [115] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.
- [116] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.
- [117] D. Ho, E. Liang, I. Stoica, P. Abbeel, and X. Chen. Population based augmentation: Efficient learning of augmentation policy schedules, 2019.
- [118] M. L. Hoffman. Empathy, social cognition, and moral action. In W. M. Kurtines and J. L. Gewirtz, editors, *Handbook of Moral Behavior and Development*, pages 1–275. L. Erlbaum, 1991.
- [119] N. N. Holland. *The dynamics of literary response*. Columbia University Press, 1989.

- [120] Q. Hu, Y. Guo, M. Cordy, X. Xie, L. Ma, M. Papadakis, and Y. Le Traon. An empirical study on data distribution-aware test selection for deep learning enhancement, Jan 2022.
- [121] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016.
- [122] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella. Taxonomy of real faults in deep learning systems. *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 1110–1121, 2020.
- [123] Y. Idelbayev. *Proper ResNet Implementation for CIFAR10/CIFAR100 in pytorch*, 2018 (accessed August 8, 2019).
- [124] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *ArXiv*, abs/1905.02175, 2019.
- [125] L. Inozemtseva and R. Holmes. Coverage is not strongly correlated with test suite effectiveness. In *Proceedings of the 36th International Conference on Software Engineering, ICSE 2014*, pages 435–445, New York, NY, USA, 2014. ACM.
- [126] International Organization for Standardization (ISO). Road vehicles – functional safety – part 6: Product development at the software level, December 2018. Second edition. Published: 2018.
- [127] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. Mixtral of experts, 2024.
- [128] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh. Pythia v0.1: the winning entry to the VQA challenge 2018. *CoRR*, abs/1807.09956, 2018.
- [129] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932, 2019.

- [130] D. R. Johnson. Transportation into a story increases empathy, prosocial behavior, and perceptual bias toward fearful expressions. *Personality and Individual Differences*, 52(2):150–155, 2012.
- [131] P. N. Johnson-Laird and K. Oatley. How poetry evokes emotions. *Acta Psychologica*, 224:103506, 2022.
- [132] K. Julian, M. Kochenderfer, and M. Owen. Deep neural network compression for aircraft collision avoidance systems. *Journal of Guidance, Control, and Dynamics*, 42:1–11, 11 2018.
- [133] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 3363–3372, 2011.
- [134] P. Kaniss. *Making local news*. University of Chicago Press, 1991.
- [135] A. Karimi, L. Rossi, and A. Prati. AEDA: an easier data augmentation technique for text classification. *CoRR*, abs/2108.13230, 2021.
- [136] A. Karimi, L. Rossi, and A. Prati. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.
- [137] A. Karimi, L. Rossi, and A. Prati. Aeda: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, 2021.
- [138] G. Kaufman and L. Libby. Changing beliefs and behavior through experience-taking. *Journal of personality and social psychology*, 103:1–19, 03 2012.
- [139] P. Ke, B. Wen, Z. Feng, X. Liu, X. Lei, J. Cheng, S. Wang, A. Zeng, Y. Dong, H. Wang, et al. Critiquellm: Towards an informative critique generation model for evaluation of large

- language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.
- [140] D. Kiela, M. Bartolo, Y. Nie, D. Kaushik, A. Geiger, Z. Wu, B. Vidgen, G. Prasad, A. Singh, P. Ringshia, et al. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, 2021.
- [141] S. Kierkegaard. *The Sickness Unto Death: The Christian Psychological Exposition for Upbuilding and Awakening*. Princeton University Press, Princeton, 1980. Originally published in 1849. Translated by H. V. Hong and E. H. Hong.
- [142] J. Kim, R. Feldt, and S. Yoo. Guiding deep learning system testing using surprise adequacy. *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1039–1049, 2019.
- [143] S. Kim, J. Suk, S. Longpre, B. Y. Lin, J. Shin, S. Welleck, G. Neubig, M. Lee, K. Lee, and M. Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics.
- [144] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [145] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.
- [146] K. Kinnick, D. Krugman, and G. Cameron. Compassion fatigue: Communication and burnout toward social problems. *Journalism & Mass Communication Quarterly - JOURNALISM MASS COMMUN*, 73:687–707, 09 1996.

- [147] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023.
- [148] J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K. Fernando, A. Saha, M. Goldblum, and T. Goldstein. On the reliability of watermarks for large language models, 2024.
- [149] J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K. Fernando, A. Saha, M. Goldblum, and T. Goldstein. On the reliability of watermarks for large language models, 2024.
- [150] M. Kiss and S. Willemsen. *Impossible Puzzle Films: A Cognitive Approach to Contemporary Complex Cinema*. Edinburgh University Press, 2017.
- [151] S. Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [152] K. Krippendorff. Computing krippendorff’s alpha-reliability, 2011.
- [153] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense, 2023.
- [154] S. Krishnan, D. Haas, M. J. Franklin, and E. Wu. Towards reliable interactive data cleaning: A user survey and recommendations. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pages 1–5, 2016.
- [155] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research), 2009.

- [156] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [157] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [158] R. Kuditipudi, J. Thickstun, T. Hashimoto, and P. Liang. Robust distortion-free watermarks for language models, 2024.
- [159] T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3075–3084, 2014.
- [160] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.
- [161] S. Kunjanman and A. Abdul Aziz. Reader-response theory: A systematic literature review. *Malaysian Journal of Social Sciences and Humanities (MJSSH)*, 6:252–260, 04 2021.
- [162] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*, pages 99–112. Chapman and Hall/CRC, 2018.
- [163] S. Kushneruk. Metaphors in the text-worlds of commercial advertising, 03 2017.
- [164] V. Lai and C. Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 29–38, 2019.
- [165] K. Lakhotia, M. Harman, and P. McMinn. A multi-objective approach to search-based test data generation, 07 2007.

- [166] S. K. Langer. *Philosophy in a new key: A study in the symbolism of reason, rite, and art*. Harvard University Press, 1942.
- [167] languagetool. languagetool, 2023.
- [168] R. Lazarus. Cognition and motivation in emotion. *The American psychologist*, 46:352–67, 04 1991.
- [169] Y. LeCun and C. Cortes. MNIST handwritten digit database. <http://yann.lecun.com/exdb/mnist/>, 2010.
- [170] J. D. Lee and K. A. See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.
- [171] M. Lee, P. Liang, and Q. Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *CHI Conference on Human Factors in Computing Systems*, CHI 22. ACM, Apr. 2022.
- [172] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019.
- [173] C. Li, J. Wang, Y. Zhang, K. Zhu, W. Hou, J. Lian, F. Luo, Q. Yang, and X. Xie. Large language models understand and can be enhanced by emotional stimuli, 2023.
- [174] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics.
- [175] J. Li, S. Ji, T. Du, B. Li, and T. Wang. Textbugger: Generating adversarial text against real-world applications. *CoRR*, abs/1812.05271, 2018.

- [176] S. Li, X. Ao, F. Pan, and Q. He. Learning policy scheduling for text augmentation. *Neural Netw.*, 145(C):121127, jan 2022.
- [177] Z. Li, X. Ma, C. Xu, and C. Cao. Structural coverage criteria for neural networks could be misleading. In *Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results*, ICSE-NIER 19, page 8992. IEEE Press, 2019.
- [178] W. Liang, G. A. Tadesse, D. Ho, L. Fei-Fei, M. Zaharia, C. Zhang, and J. Zou. Advances, challenges and opportunities in creating data for trustworthy AI. *Nature Machine Intelligence*, 4(8):669–677, 2022.
- [179] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim. Fast autoaugment, 2019.
- [180] B. Y. Lin, M. Shen, Y. Xing, P. Zhou, and X. Ren. Commongen: A constrained text generation dataset towards generative commonsense reasoning. *CoRR*, abs/1911.03705, 2019.
- [181] S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. *CoRR*, abs/2109.07958, 2021.
- [182] A. Liu, L. Pan, X. Hu, S. Meng, and L. Wen. A semantic invariant robust watermark for large language models, 2024.
- [183] A. Liu, L. Pan, Y. Lu, J. Li, X. Hu, X. Zhang, L. Wen, I. King, H. Xiong, and P. Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024.
- [184] C. Y. Liu, L. Zeng, J. Liu, R. Yan, J. He, C. Wang, S. Yan, Y. Liu, and Y. Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.
- [185] X. Liu, L. Zhang, and S. Hong. Global biodiversity research during 19002009: a bibliometric analysis. *Biodiversity and Conservation*, 20:807–826, 2010.
- [186] Y. Liu and Y. Bu. Adaptive text watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org, 2024.

- [187] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, Dec. 2023. Association for Computational Linguistics.
- [188] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [189] Y. Liu, Y. Su, E. Shareghi, and N. Collier. Unlocking structure measuring: Introducing PDD, an automatic metric for positional discourse coherence. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 92–100, Mexico City, Mexico, June 2024. Association for Computational Linguistics.
- [190] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [191] S. Longpre, R. Mahari, N. Obeng-Marnu, W. Brannon, T. South, J. Kabbara, and S. Pentland. Data authenticity, consent, and provenance for ai are all broken: What will it take to fix them?, 2024.
- [192] J. Lou. The relation between evenness and diversity. *Diversity*, 2, 02 2010.
- [193] S. Lundberg et al. Guidance. <https://github.com/guidance-ai/guidance>, 2022.
- [194] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, and Y. Wang. Deepgauge: Multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, ASE 2018, pages 120–131, New York, NY, USA, 2018. ACM.

- [195] W. Ma, M. Papadakis, A. Tsakmalis, M. Cordy, and Y. L. Traon. Test selection for deep learning systems. *ArXiv*, abs/1904.13195, 2019.
- [196] Y. Ma, K. L. Nguyen, F. Z. Xing, and E. Cambria. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70, 2020.
- [197] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
- [198] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
- [199] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.
- [200] S. Maier, P. Slovic, and M. Mayorga. Reader reaction to news of mass suffering: Assessing the influence of story form and emotional response. *Journalism*, 18, 08 2016.
- [201] S. R. Maier. Compassion fatigue and the elusive quest for journalistic impact: A content and reader-metrics analysis assessing audience response. *Journalism & Mass Communication Quarterly*, 92(3):700–722, 2015.
- [202] S. Mailloux. Stanley fish’s "interpreting the variorum": Advance or retreat? *Critical Inquiry*, 3(1):183–190, 1976.
- [203] R. Mar, K. Oatley, M. Djikic, and J. Mullin. Emotion and narrative fiction: Interactive influences before, during, and after reading. *Cognition & emotion*, 25:818–33, 08 2011.
- [204] R. A. Mar, K. Oatley, J. Hirsh, J. Dela Paz, and J. B. Peterson. Bookworms versus nerds: Exposure to fiction versus non-fiction, divergent associations with social ability, and the

- simulation of fictional social worlds. *Journal of research in personality*, 40(5):694–712, 2006.
- [205] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.
- [206] W. M. McKeeman. Differential testing for software. *Digital Technical Journal*, 10(1):100–107, 1998.
- [207] O. Megalaki, U. Ballenghein, and T. Baccino. Effects of valence and emotional intensity on the comprehension and memorization of texts. *Frontiers in Psychology*, 10, 2019.
- [208] W. B. Mendes. Handbook of emotions, fourth edition, 2016.
- [209] L. Meng, C.-T. Lin, T.-P. Jung, and D. Wu. White-box target attack for eeg-based bci regression problems. In *International Conference on Neural Information Processing*, pages 476–488. Springer, 2019.
- [210] D. S. Miall and D. Kuiken. Shifting perspectives: Readers feelings and literary response. *New perspectives on narrative perspective*, pages 289–301, 2001.
- [211] P. Minervini and S. Riedel. Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics.
- [212] Y. Minghao. infly/INF-ORM-Llama3.1-70B @ Hugging Face — huggingface.co. <https://huggingface.co/infly/INF-ORM-Llama3.1-70B>, 2024. [Accessed 11-02-2025].
- [213] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online, Oct. 2020. Association for Computational Linguistics.

- [214] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020.
- [215] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. *arXiv preprint arXiv:2005.05909*, 2020.
- [216] mseitzer. *Frèchet Inception Distance (FID score) in PyTorch*, 2018 (accessed August 9, 2019).
- [217] A. Mumuni and F. Mumuni. Data augmentation: A comprehensive survey of modern approaches. *Array*, 16:100258, 2022.
- [218] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, and D. Woodard. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSuR)*, 50(6):1–36, 2017.
- [219] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, et al. Reading digits in natural images with unsupervised feature learning, 2011.
- [220] G. E. Newman and R. K. Smith. Kinds of authenticity. *Philosophy Compass*, 11:609–618, 2016.
- [221] A. M. Nguyen, J. Yosinski, and J. Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *ArXiv*, abs/1602.03616, 2016.
- [222] T. Niu and M. Bansal. Automatically learning data augmentation policies for dialogue tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1317–1323, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.
- [223] C. G. Northcutt, L. Jiang, and I. L. Chuang. Confident learning: Estimating uncertainty in dataset labels, 2022.

- [224] M. Nussbaum. Emotions as judgments of value and importance. In R. C. Solomon, editor, *Thinking About Feeling: Contemporary Philosophers on Emotions*. Oxford University Press, 2004.
- [225] K. Oatley. Emotions and the story worlds of fiction. *Narrative Impact: Social and Cognitive Foundations*, pages 39–69, 01 2002.
- [226] A. Odena and I. J. Goodfellow. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. In *International Conference on Machine Learning*, 2019.
- [227] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. <https://distill.pub/2017/feature-visualization>.
- [228] OpenAI. Gpt-4 turbo, Nov. 2024. Accessed: 2025-02-10.
- [229] OpenAI. Gpt-4o system card, May 2024. Accessed: 2025-02-10.
- [230] OpenAI. Openai o3-mini, 2025. Accessed: 2025-02-10.
- [231] A. Ortony, G. L. Clore, and A. Collins. *The cognitive structure of emotions*. Cambridge university press, 2022.
- [232] A. Paleyes, R.-G. Urma, and N. D. Lawrence. Challenges in deploying machine learning: a survey of case studies. *ACM Computing Surveys*, 55(6):1–29, 2022.
- [233] L. Pan, A. Liu, Z. He, Z. Gao, X. Zhao, Y. Lu, B. Zhou, S. Liu, X. Hu, L. Wen, I. King, and P. S. Yu. MarkLLM: An open-source toolkit for LLM watermarking. In D. I. Hernandez Farias, T. Hope, and M. Li, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–71, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics.
- [234] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

- [235] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In P. Isabelle, E. Charniak, and D. Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
- [236] J. Park, S. Jwa, M. Ren, D. Kim, and S. Choi. Offsetbias: Leveraging debiased data for tuning evaluators, 2024.
- [237] M. R. Parvez, T. Bolukbasi, K. Chang, and V. Sarigrama. Building a robust text classifier on a test-time budget. *CoRR*, abs/1808.08270, 2018.
- [238] K. Pei, Y. Cao, J. Yang, and S. Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP ’17, pages 1–18, New York, NY, USA, 2017. ACM.
- [239] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017.
- [240] J. Petraglia. The importance of being authentic: Persuasion, narration, and dialogue in health communication and education. *Health communication*, 24(2):176–185, 2009.
- [241] H. Pham, X. Wang, Y. Yang, and G. Neubig. Meta back-translation, 2021.
- [242] H. Phan. *PyTorch models trained on CIFAR-10 dataset*, 2019 (accessed October 4, 2019).
- [243] E. C. Pielou. The measurement of diversity in different types of biological collections. *Journal of theoretical biology*, 13:131–144, 1966.
- [244] C. Pierse. Transformers Interpret, 2 2021.
- [245] Y. Pruksachatkun, S. Krishna, J. Dhamala, R. Gupta, and K. Chang. Does robustness improve fairness? approaching fairness with word substitution robustness methods for text classification. *CoRR*, abs/2106.10826, 2021.
- [246] R. Pryzant, D. Iter, J. Li, Y. Lee, C. Zhu, and M. Zeng. Automatic prompt optimization with “gradient descent” and beam search. In H. Bouamor, J. Pino, and K. Bali, editors,

- Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore, Dec. 2023. Association for Computational Linguistics.
- [247] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. When training and test sets are different: Characterizing learning transfer, 2009.
- [248] H. Quteineh, S. Samothrakis, and R. Sutcliffe. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7400–7410, Online, Nov. 2020. Association for Computational Linguistics.
- [249] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019.
- [250] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [251] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.
- [252] N. Rajani, N. Lambert, S. Han, J. Wang, O. Nitski, E. Beeching, L. Turnstall, J. Launay, and T. Wolf. Can foundation models label data like humans? *Hugging Face Blog*, 2023. <https://huggingface.co/blog/llm-leaderboard>.
- [253] V. Raman and J. M. Hellerstein. Potter’s wheel: An interactive data cleaning system. In *VLDB*, volume 1, pages 381–390, 2001.
- [254] S. Rastogi and D. Pruthi. Revisiting the robustness of watermarking to paraphrasing attacks. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18100–18110, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics.
- [255] B. Ratner. The correlation coefficient: Its values range between +1/-1, or do they? *Journal of Targeting, Measurement and Analysis for Marketing*, 17, 06 2009.

- [256] B. Ratner. The correlation coefficient: Its values range between +1/1, or do they? *Journal of Targeting, Measurement and Analysis for Marketing*, 17(2):139–142, Jun 2009.
- [257] J. Ren, H. Xu, Y. Liu, Y. Cui, S. Wang, D. Yin, and J. Tang. A robust semantics-based watermark for large language model against paraphrasing, 2024.
- [258] S. Ren, J. Zhang, L. Li, X. Sun, and J. Zhou. Text AutoAugment: Learning compositional augmentation policy for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9029–9043, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.
- [259] E. K. Rezig, L. Cao, M. Stonebraker, G. Simonini, W. Tao, S. Madden, M. Ouzzani, N. Tang, and A. K. Elmagarmid. Data civilizer 2.0: A holistic framework for data preparation and analytics. *Proceedings of the VLDB Endowment*, 12(12):1954–1957, 2019.
- [260] M. T. Ribeiro and S. Lundberg. Adaptive testing and debugging of NLP models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3253–3267, 2022.
- [261] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics.
- [262] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *ACL*, 2020.
- [263] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, 2020.
- [264] J.-J. Rousseau. A discourse on the origin of inequality. In *Democracy: A Reader*, pages 127–129. Columbia University Press, 2016. Originally published in 1755.

- [265] S. Sabour, C. Zheng, and M. Huang. Cem: Commonsense-aware empathetic response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11229–11237, 2022.
- [266] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.
- [267] L. Salewski, S. Alaniz, I. Rio-Torto, E. Schulz, and Z. Akata. In-context impersonation reveals large language models' strengths and biases. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 72044–72057. Curran Associates, Inc., 2023.
- [268] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *ArXiv*, abs/1606.03498, 2016.
- [269] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo. everyone wants to do the model work, not the data work: Data cascades in high-stakes AI. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.
- [270] J. Sartre. *Sketch for a Theory of the Emotions*. Routledge, 1971. Originally published in 1939.
- [271] L. M. Scott. The Bridge from Text to Mind: Adapting Reader-Response Theory to Consumer Research. *Journal of Consumer Research*, 21(3):461–480, 12 1994.
- [272] S. Segura, G. Fraser, A. B. Sánchez, and A. Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on Software Engineering*, 42:1–1, 09 2016.
- [273] J. Sekhon and C. Fleming. Towards improved testing for deep learning. *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pages 85–88, 2019.
- [274] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, Jan. 2001.

- [275] L. Shi. Trading-off among accuracy, similarity, diversity, and long-tail: a graph-based recommendation approach. *Proceedings of the 7th ACM conference on Recommender systems*, 2013.
- [276] C. Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019.
- [277] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [278] P. Simard, Y. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition-tangent distance and tangent propagation. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, page 23927, Berlin, Heidelberg, 1998. Springer-Verlag.
- [279] T. Singer, B. Seymour, J. O’Doherty, K. Stephan, R. Dolan, and C. Frith. Empathic neural responses are modulated by the perceived fairness of others. *Nature*, 439:466–9, 02 2006.
- [280] P. Slovic. If i look at the mass i will never act: Psychic numbingpsychic numbing and genocidegenocide. *Judgment and Decision Making*, 2:79–95, 02 2007.
- [281] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics.
- [282] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics.
- [283] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the*

- 2013 conference on empirical methods in natural language processing, pages 1631–1642, 2013.
- [284] S. Somasundaran, M. Flor, M. Chodorow, H. Molloy, B. Gyawali, and L. McCulla. Towards evaluating narrative quality in student writing. *Transactions of the Association for Computational Linguistics*, 6:91–106, 2018.
- [285] A. Spangher, N. Peng, S. Gehrmann, and M. Dredze. Do llms plan like human writers? comparing journalist coverage of press releases with llms. In *Conference on Empirical Methods in Natural Language Processing*, 2024.
- [286] N. Speer, J. Reynolds, K. Swallow, and J. Zacks. Reading stories activates neural representations of visual and motor experiences. *Psychological science*, 20:989–99, 07 2009.
- [287] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
- [288] G. Steiner. On difficulty. *Journal of Aesthetics and Art Criticism*, 36(3):263–276, 1978.
- [289] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize from human feedback. *CoRR*, abs/2009.01325, 2020.
- [290] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *ArXiv*, abs/1703.01365, 2017.
- [291] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2015.
- [292] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.
- [293] C. Taylor. *The Ethics of Authenticity*. Harvard University Press, Cambridge, 1991.

- [294] L. Tian and H. Wang. Towards a text-world approach to translation and its pedagogical implications. *International Journal of Translation, Interpretation, and Applied Linguistics*, 1:14–26, 01 2019.
- [295] Y. Tian, T. Huang, M. Liu, D. Jiang, A. Spangher, M. Chen, J. May, and N. Peng. Are large language models capable of generating human-level narratives?, 2024.
- [296] Y. Tian, K. Pei, S. Jana, and B. Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering, ICSE '18*, pages 303–314, New York, NY, USA, 2018. ACM.
- [297] P. Tikka and M. Kaipainen. Intersubjectivity, idiosyncrasy and narrative deixis a neurocinematic approach, 2017.
- [298] L. Torgo and J. Gama. Regression by classification. In *Advances in Artificial Intelligence: 13th Brazilian Symposium on Artificial Intelligence, SBIA'96 Curitiba, Brazil, October 23–25, 1996 Proceedings 13*, pages 51–60. Springer, 1996.
- [299] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [300] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova. Well-read students learn better: On the importance of pre-training compact models, 2019.
- [301] UCLA-SEAL. UCLA-SEAL/ProvenanceInspector. <https://github.com/UCLA-SEAL/ProvenanceInspector>, 2023.
- [302] Udacity. Self-Driving Car Engineer Nanodegree Program Code. GitHub Repository, 2025. Accessed: 2019-09-01.
- [303] L. Q. Uddin, M. Iacoboni, C. Lange, and J. P. Keenan. The self and social cognition: the role of cortical midline structures and mirror neurons. *Trends in Cognitive Sciences*, 11(4):153–157, 2007.

- [304] M. Vaezi and S. Rezaei. Development of a rubric for evaluating creative writing: a multi-phase research. *New Writing*, 16(3):303–317, 2019.
- [305] V. Verma, E. Fleisig, N. Tomlin, and D. Klein. Ghostbuster: Detecting text ghostwritten by large language models. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico, June 2024. Association for Computational Linguistics.
- [306] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics.
- [307] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- [308] D. Wang, S. Prabhat, and N. Sambasivan. Whose AI dream? in search of the aspiration in data annotation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2022.
- [309] H. Wang, W. Xiong, T. Xie, H. Zhao, and T. Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *EMNLP*, 2024.
- [310] J. Wang, D. Crawl, S. Purawat, M. Nguyen, and I. Altintas. Big data provenance: Challenges, state of the art and opportunities. In *2015 IEEE international conference on big data (Big Data)*, pages 2509–2516. IEEE, 2015.
- [311] S. Wang, Y. Liu, Y. Xu, C. Zhu, and M. Zeng. Want to reduce labeling cost? GPT-3 can help. *arXiv preprint arXiv:2108.13487*, 2021.

- [312] T. Wang, J. Chen, Q. Jia, S. Wang, R. Fang, H. Wang, Z. Gao, C. Xie, C. Xu, J. Dai, et al. Weaver: Foundation models for creative writing, 2024.
- [313] A. Warstadt, A. Singh, and S. R. Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.
- [314] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, Q. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022.
- [315] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
- [316] J. Wei and K. Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.
- [317] J. Wei and K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.
- [318] J. W. Wei and K. Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196, 2019.
- [319] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, and H. Xu. Time series data augmentation for deep learning: A survey. *CoRR*, abs/2002.12478, 2020.
- [320] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics*, 26(1):56–65, 2019.
- [321] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-Art Natural Language Processing, Oct. 2020.

- [322] A. Woodruff and R. Griffin. Reader response in secondary settings: Increasing comprehension through meaningful interactions with literary texts. *Texas Journal of Literacy Education*, 5, 01 2017.
- [323] T. Wu, K. Wongsuphasawat, D. Ren, K. Patel, and C. DuBois. Tempura: Query analysis with structural templates. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.
- [324] Q. Xie, Z. Dai, E. H. Hovy, M. Luong, and Q. V. Le. Unsupervised data augmentation. *CoRR*, abs/1904.12848, 2019.
- [325] J. xiong Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.
- [326] G. Xu, R. Liu, F. Harel-Canada, N. R. Chandra, and N. Peng. EnDex: Evaluation of dialogue engagingness at scale. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4884–4893, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.
- [327] S. Yan, G. Tao, X. Liu, J. Zhai, S. Ma, L. Xu, and X. Zhang. Correlations between deep neural network model coverage criteria and model quality. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 775–787, 2020.
- [328] K. Yang, Y. Tian, N. Peng, and D. Klein. Re3: Generating longer stories with recursive reprompting and revision. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.
- [329] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

- [330] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.
- [331] Z. Yang, R. O. Sinnott, J. Bailey, and Q. Ke. A survey of automated data augmentation algorithms for deep learning-based image classification tasks, 2022.
- [332] L. Yao, N. Peng, R. Weischedel, K. Knight, D. Zhao, and R. Yan. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385, 2019.
- [333] M. Ye, C. Gong, and Q. Liu. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. *CoRR*, abs/2005.14424, 2020.
- [334] S. Yoon, G. Kim, and K. Park. SSMix: Saliency-based span mixup for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3225–3234, Online, Aug. 2021. Association for Computational Linguistics.
- [335] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *ArXiv*, abs/1506.06579, 2015.
- [336] A. Yuan, A. Coenen, E. Reif, and D. Ippolito. Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces, IUI '22*, page 841852, New York, NY, USA, 2022. Association for Computing Machinery.
- [337] Y. Yuan, Q. Pang, and S. Wang. Revisiting neuron coverage for dnn testing: A layer-wise and distribution-aware criterion. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 1200–1212. IEEE, 2023.
- [338] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *CoRR*, abs/1905.04899, 2019.
- [339] J. Zaki and K. Ochsner. Zaki j, ochsner k. the neuroscience of empathy: progress, pitfalls and promise. *nat neurosci* 15: 675-680. *Nature neuroscience*, 15:675–80, 04 2012.

- [340] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017.
- [341] H. Zhang, B. L. Edelman, D. Francati, D. Venturi, G. Ateniese, and B. Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. In *Forty-first International Conference on Machine Learning*, 2024.
- [342] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. Machine learning testing: Survey, landscapes and horizons. *CoRR*, abs/1906.10742, 2019.
- [343] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. Machine learning testing: Survey, landscapes and horizons. *CoRR*, abs/1906.10742, 2019.
- [344] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018*, pages 132–142, New York, NY, USA, 2018. ACM.
- [345] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019.
- [346] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3), apr 2020.
- [347] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification, 2016.
- [348] X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626, 2015.
- [349] Z. Zhao, S. Song, B. Duah, J. Macbeth, S. Carter, M. P. Van, N. S. Bravo, M. Klenk, K. Sick, and A. L. S. Filipowicz. More human than human: Llm-generated narratives outperform human-llm interleaved narratives. In *Proceedings of the 15th Conference on Creativity and*

- Cognition*, C&C '23, page 368370, New York, NY, USA, 2023. Association for Computing Machinery.
- [350] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. *CoRR*, abs/1708.04896, 2017.
 - [351] F. Zhou, J. Li, C. Xie, F. Chen, L. Hong, R. Sun, and Z. Li. Metaaugment: Sample-aware data augmentation policy learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11097–11105, 2021.
 - [352] J. Zhou, Y. Zhang, Q. Luo, A. G. Parker, and M. De Choudhury. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2023.
 - [353] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba. Large language models are human-level prompt engineers, 2023.
 - [354] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.
 - [355] L. Zunshine. Why we read fiction: Theory of mind and the novel. *Bibliovault OAI Repository, the University of Chicago Press*, 01 2006.
 - [356] L. Zwarun and A. Hall. Narrative persuasion, transportation, and the role of need for cognition in online viewing of fantastical films. *Media Psychology*, 15, 07 2012.