

CLEAR: Contrasting Textual Feedback with Experts and Amateurs for Reasoning

Anonymous ACL submission

Abstract

We introduce CLEAR (Contrasting Textual Feedback with Experts and Amateurs for Reasoning), a novel approach to language model reasoning that leverages the strengths of a larger (expert) model and smaller (amateur) model. The expert and amateur models each provide feedback on a model’s initial output and are contrasted with each other into refined feedback. This feedback is subsequently applied to iteratively improve CLEAR’s responses. Our experiments demonstrate that CLEAR outperforms state-of-the-art methods in several challenging reasoning tasks, including story outline improvement (up to 19.6% relative increase in interestingness), constrained generation (up to 18.5% increase in coverage), mathematical reasoning (up to 6.7% improvement in accuracy) and mitigation of toxicity (decrease of up to 22% in toxicity).

1 Introduction

Large Language Models (LLMs) such as GPT (Brown et al., 2020; OpenAI, 2024), LLaMA (Touvron et al., 2023a,b), and Claude (Anthropic, 2024) have shown increasing reasoning capabilities with certain prompting techniques. Despite these advances, many methods that incorporate feedback in the reasoning process do not include sufficient mechanisms to verify the feedback’s quality and accuracy, making it challenging to consistently improve model outputs.

Existing prompting techniques like Chain-of-Thought (CoT) (Wei et al., 2022) generate an output using intermediate steps that are termed "chains of thought". Self-consistency (SC) (Wang et al., 2022) produces multiple chains of thought and selects the most consistent and repeated outcome. Newer methods such as Tree-of-Thoughts (ToT) (Yao et al., 2023), Graph-of-Thoughts (GoT) (Besta et al., 2024)

and THOUGHTSCULPT (Chi et al., 2024) utilize a graphical tree structure, enabling the exploration of multiple reasoning paths and revision steps. However, these tree-based methods suffer from computational overheads. Additionally, errors in reasoning or feedback propagate through iterations.

For these reasons, we propose CLEAR, a novel framework that provides precise feedback on a model’s output to further refine it. CLEAR is significantly more computationally efficient than other tree-based methods (see Appendix C). Similar to how humans would contrast and incorporate multiple feedback they receive to form a high-quality evaluation (Mamad, 2023), our method contrasts expert and amateur model feedback. In this case, the expert is a larger LM and the amateur is a smaller LM. This allows the model to receive a holistic review of the entire output at once, enabling CLEAR to be deployed for any task. The main advantage of employing models with different sizes in CLEAR lies in their ability to diversify the feedback while remaining cost-effective (see Appendix C). In addition, contrasting different models may reduce hallucinations and prevent inaccuracies from accumulating over the iterations as discussed in Shi et al. (2023).

The process of providing feedback can be repeated with the improved output for further refinement, creating a graphical structure with nodes representing each output as shown in Figure 1. We also propose the use of a best-first search algorithm as a pathfinding algorithm for tasks with objective solutions. We use a search algorithm to expand the most promising node instead of the last node generated, allowing the improvement to be done on the better nodes which leads to better final results.

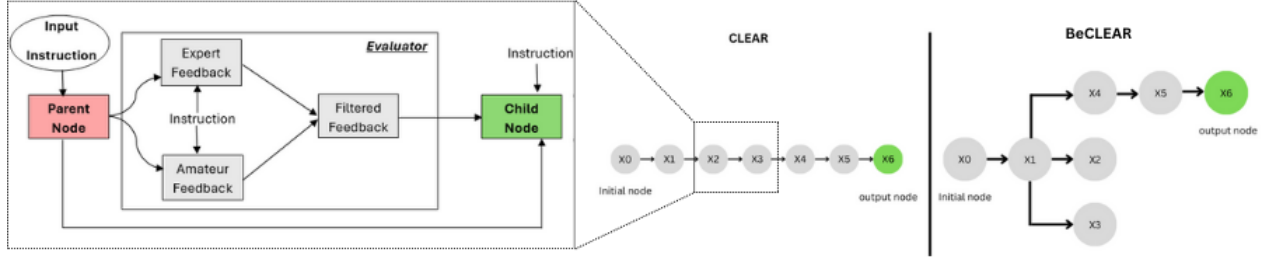


Figure 1: This diagram demonstrated the two variants of CLEAR and shows how the best-first search is leveraged to improve the most promising nodes only.

2 Methods

Research has shown that when humans are in the learning process, they rely heavily on high-quality feedback from other people (Mamad, 2023); however, this is not always accessible for LLMs, so multiple feedback sources can be beneficial (Yamagata et al., 2021). Therefore, CLEAR’s approach uses two LLMs of different sizes to provide feedback. Importantly, the two feedbacks are contrasted to create more efficient advice. We propose a method to achieve this primarily through the use of prompts.

It is standard to use p_θ to denote a pre-trained LM with parameters θ and other lowercase letters like x, y, z, \dots to denote language sequences. For example, $x = (x^0, x^1, \dots, x^i)$ where x^n is a token such that $p_\theta(x) = \prod_{i=1}^n p_\theta(x^i | x^{1 \dots i})$. Additionally, the previous outputs and their feedback are provided in those prompts to help guide the model in improving the new output more accurately (see Appendix A). This allows CLEAR to be quickly implemented in various tasks.

This paper will treat each output of the LM as a node $x \in \{x^0, x^1, \dots, x^i\}$, where x^0 is the root node and the initial output provided by the model given an instruction I . Each node represents a full output of the LLM and stores all of the feedback received. Furthermore, each node stores the expert, amateur, and filtered feedback it received. To implement CLEAR, two modules are required to handle and generate feedback: Node Evaluator, and Feedback filter. These modules can be implemented with only three prompts. The Node Evaluator generates the expert and amateur evaluations, while

the Feedback filter contrasts them to produce the "filtered feedback".

2.1 Receiving feedback

Node Evaluator. The Node Evaluator provides a holistic review of each output node x according to the instruction I . The feedback $f_{\text{textual}}(x^i)$ consists of a textual evaluation. $f_{\text{textual}}(x^i)$ provides feedback containing all the positive and negative aspects of the node, as well as possible areas of improvement if applicable. For each task, the feedback prompt is slightly changed to address the problem more effectively (see Appendix A). This process is done with the expert and amateur models to produce $f_{\text{textual}}^{\text{expert}}(x^i)$, $f_{\text{textual}}^{\text{amateur}}(x^i)$ respectively.

Feedback Filter. Recognizing that LLMs often produce factual inaccuracies, especially in subjective tasks like feedback generation, we use an LLM to process and contrast the expert and amateur feedback creating $f_{\text{filtered}}(x^i)$ (see Figure 2). A higher priority is given to the expert’s input as done in (O’Brien and Lewis, 2023).

$$f_{\text{filtered}}(x^i) \sim p_\theta(f_{\text{textual}}^{\text{amateur}}(x^i), f_{\text{textual}}^{\text{expert}}(x^i)) \quad (1)$$

We utilize an LLM for this module since prompting it to contrast the two feedback is sufficient to produce high-quality filtered feedback, allowing CLEAR to be deployed using prompts only.

After obtaining the filtered feedback $f_{\text{filtered}}(x^i)$ for the parent node x_{parent} according to the instruction I , we can get the improved child node x_{child} . This process can be repeated with each child node

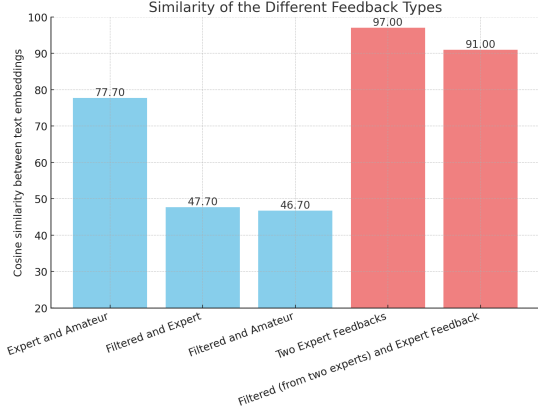


Figure 2: Cosine similarity analysis between different feedback types using text-embeddings-3-small. These results are the average similarities aggregated across 200 data points in the GSM8K and CommonGen-Hard experiments. The bar graph shows that the expert and amateur model feedback are semantically different, and the filtered feedback also contains different content. Furthermore, if two of the same models are used instead (red bars), the filtered feedback does not contain significantly different content, worsening CLEAR’s performance.

to produce better results. We denote each iteration (and consequently the number of improvement cycles) as d , where $d=1$ represents one improvement of the output.

$$x_{child} \sim p_{\theta}(x|I, x_{parent}, f_{filtered}(x_{parent})) \quad (2)$$

Algorithm 1 CLEAR($I, p_{\theta}, q_{\theta}, d$)

Require: Instruction I , pretrained expert LLM p_{θ} , pretrained amateur LLM q_{θ} , number of iterations d

- 1: **Input:** initial node x_0
- 2: **Output:** final node x_d
- 3: **for** $j = 1$ to d **do**
- 4: $Expert \leftarrow \text{EVALUATE}(x_{parent}, p_{\theta}, I)$
- 5: $Amateur \leftarrow \text{EVALUATE}(x_{parent}, q_{\theta}, I)$
- 6: $Filtered \leftarrow \text{CONTRAST}(Amateur, Expert)$
- 7: Expand parent node x_{parent} with a new child node x_j
- 8: $x_j \leftarrow \text{GENERATECHILD}(I, x_{parent}, Filtered, p_{\theta})$
- 9: **end for**
- 10: **return** x_d

2.2 Search Algorithms

Usually in CLEAR, each x_{parent} is linked to a single x_{child} (Figure 1). However, for reasoning tasks where the final answer is objective, such as in mathematical questions, deploying a best-first search algorithm with CLEAR, which we will call BeCLEAR, is more effective. Other search algorithms such as A*, DFS, and BFS can be

alternatives; however, this paper does not test them.

Best-First Search. For best-first search, we request the expert and amateur feedback to include a numerical score of the model output $v(expert)$ and $v(amateur)$. The expert score for the root node will be denoted as v_0 .

BeCLEAR aims to go from the initial output to the ideal response which is assumed to receive a score of 100. Since each node has an infinite number of potential neighbors, we use a cost function $g(n)$ [3] and heuristic $h(n)$ [4] between the nodes which are summed to find the total cost $f(n) = g(n) + h(n)$. Finally, the "best" node having the lowest $f(n)$ is explored first (see Figure 1).

$$g(n) = |v_0 - v(expert)| + |v_0 - v(amateur)| \quad (3)$$

$$h(n) = 100 - |v(expert) - v(amateur)| \quad (4)$$

Algorithm 2 BeCLEAR($I, p_{\theta}, q_{\theta}, d$)

Require: Instruction I , pretrained expert LLM p_{θ} , pretrained amateur LLM q_{θ} , number of iterations d

- 1: **Input:** initial node x_0
- 2: **Output:** final node x_d
- 3: $S \leftarrow \{x_0\}$ ▷ Initialize set of nodes
- 4: **for** $j = 1$ to d **do**
- 5: $Expert \leftarrow \text{EVALUATE}(x_{parent}, p_{\theta}, I)$
- 6: $Amateur \leftarrow \text{EVALUATE}(x_{parent}, q_{\theta}, I)$
- 7: $Filtered \leftarrow \text{CONTRAST}(Amateur, Expert)$
- 8: Select node $x_{min} \in S$ with lowest $f(n)$
- 9: Expand parent node x_{min} with a new child node x_j
- 10: $x_j \leftarrow \text{GENERATECHILD}(I, x_{parent}, Filtered, p_{\theta})$
- 11: $S \leftarrow S \cup \{x_j\}$ ▷ Add new node to set
- 12: **end for**
- 13: **return** x_d

3 Experiments and Results

We evaluate CLEAR, with GPT-4o as the expert model and GPT-3.5-turbo as the amateur model, on four difficult tasks. We have conducted mathematical reasoning experiments using the **GSM8K** dataset (Cobbe et al., 2021) and the **agieval-math** dataset (Zhong et al., 2023), constrained generation using the **CommonGen-Hard** dataset (Madaan et al., 2024), story outline improvement using **WhatsThatBook** dataset (Lin et al., 2023), and toxicity mitigation using **RealToxicityMitigation** (Gehman et al., 2020). Each experiment evaluates CLEAR and other methods on essential reasoning skills since they require lexical, informative, mathematical, and commonsense abilities. We

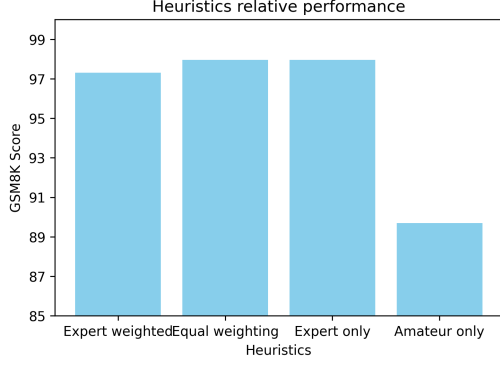


Figure 3: Different heuristics for the best-first search were tested on GSM8K with $d = 5$ (see section 3.3). Expert weighted: $100 - |1.5v(\text{expert}) - v(\text{amateur})|$, Equal weighting: $100 - |v(\text{expert}) - v(\text{amateur})|$, Expert only: $100 - v(\text{expert})$, Amateur only: $100 - v(\text{amateur})$.

will denote each iteration of CLEAR as d . For Tree of Thoughts and THOUGHTSCULPT, d will represent the maximum node depth. Our main results use $d=3$, but we have also tested CLEAR for $d \in \{1, 2, 3, 4, 5\}$ and found that each further iteration improves the model.

3.1 Constrained Generation

Task setup. We use CommonGen-Hard, a benchmarking dataset used to evaluate the commonsense abilities in LLM text generation. CommonGen-Hard, which encompasses 20-30 concepts, was introduced in (Madaan et al., 2024) as a harder variant to CommonGen (Lin et al., 2020), which only uses four concepts.

Method setup. We use GPT-4o as the base model for the LLM. All the methods will use $d=3$.

Methods	Coverage(%)
CoT	96.1
ToT	98.8
THOUGHTSCULPT (DFS)	99.1
THOUGHTSCULPT (MCTS)	99.0
CLEAR	99.3
BeCLEAR (d=2)	97.5
BeCLEAR (d=3)	97.0

Table 1: CommonGen-Hard percentage coverage results with $d=3$.

Results. Table 1 shows that CLEAR covers the most concepts with 99.3%. Since a model can simply maximize the concept coverage without making the sentences logical, we opted to benchmark

CLEAR on how the ideas were utilized. For this, we deployed GPT-4o to rate each sentence according to relevance and comprehensibility of concepts and sentences. Figure 5 shows CLEAR’s scores for different values of d .

3.2 Story Outline Improvement

Previous research (Yang et al., 2022, 2023) demonstrates that generating a high-level outline for a story first is beneficial. Therefore, we propose a story outline improvement task.

Task setup. We use the WhatsThatBook dataset (Lin et al., 2023) to sample 500 book descriptions as done in THOUGHTSCULPT (Chi et al., 2024). The book descriptions are used to create story outlines using DOC (Yang et al., 2022) with GPT-3.5 as the base model. For this test, we specifically focus on the level of interestingness and creativity relative to the default outlines.

Method setup. GPT-4o is used as the base model. All the methods will use $d=3$. GPT-4o was used as the content evaluator.

Methods	Interestingness(%)
Initial Outline	12
CoT	35
ToT	43
ThoughSculpt (MCTS)	61
CLEAR	73

Table 2: Average percentage of the outline’s interestingness. The initial outline is the original outline that got improved by the other methods. Relative to the other methods, CLEAR produces the most interesting outlines.

Results. Table 2 shows the results for each method. In this experiment, it is obvious that each method substantially improves the interestingness of the outline. However, CLEAR achieves the highest result of 73.1%.

3.3 Mathematical Reasoning

Task setup. We use GSM8K (Cobbe et al., 2021) which contains grade level math word problems, and the **agieval-math** dataset (Zhong et al., 2023) which contains more difficult questions, which some LLMs struggle with.

Method setup. We use GPT-4o as the base model for the LLM. We will also use zero-shot prompting.

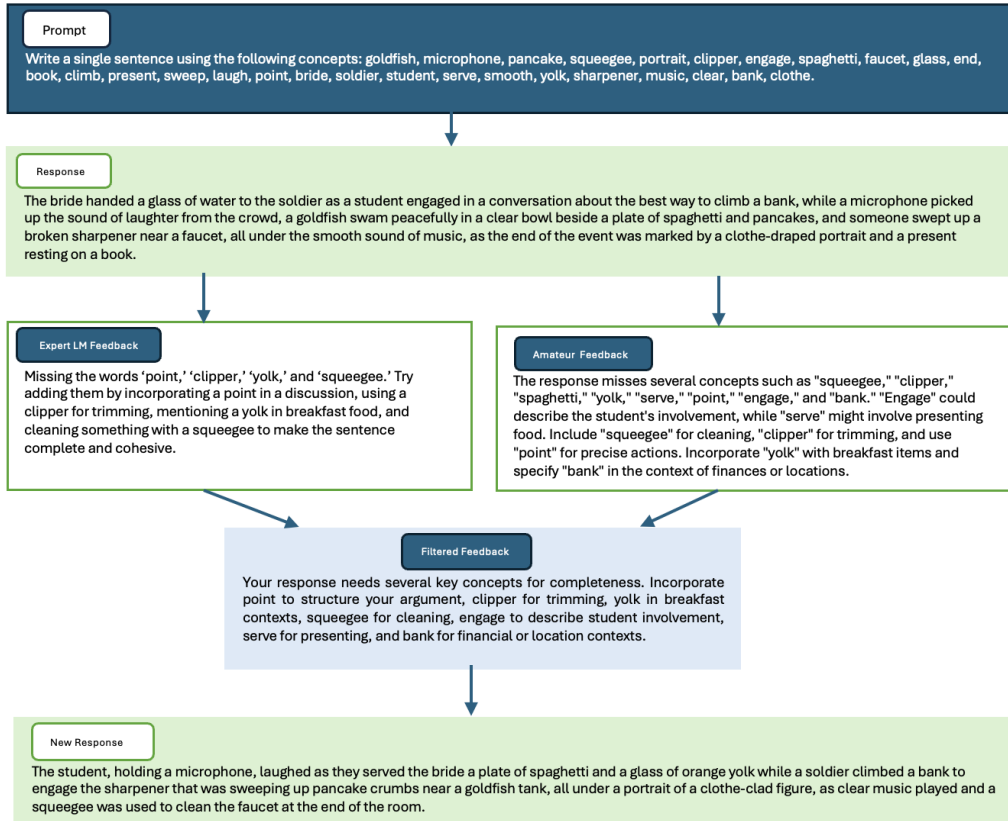


Figure 4: In this diagram, we demonstrate that the expert and amateur models’ feedback are processed in constrained generation.

Methods	GSM8K(%)	agieval-math
CoT	86.0	73.3
ToT	90.0	-
CLEAR (d=2)	91.0	-
THOUGHTSCULPT	96.0	66.0
CLEAR (d=3)	96.8	78.1
BeCLEAR (d=2)	96.9	-
BeCLEAR (d=3)	97.0	78.2
BeCLEAR (d=4)	97.2	-

Table 3: The accuracies for GSM8K and agieval-math in percentages. CLEAR has the highest scores across the board.

Results. Table 3 provides the results of the tests. As expected, all the methods achieve higher scores than Chain-of-Thought, but BeCLEAR outperforms other methods, with BeCLEAR (d=4) only missing 2.8% of GSM8K’s questions.

3.4 Toxicity mitigation

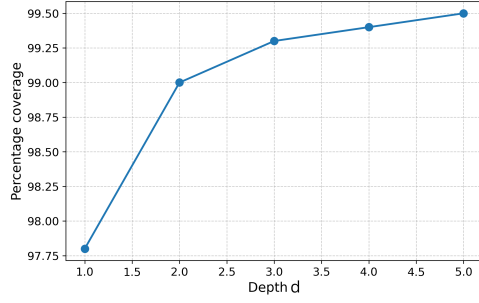
Considering the importance for LLMs to produce non-toxic text for ethical reasons, we have tested if CLEAR can mitigate harmful generations.

Task Setup. We use the RealToxicityPrompts dataset (Gehman et al., 2020) which is designed

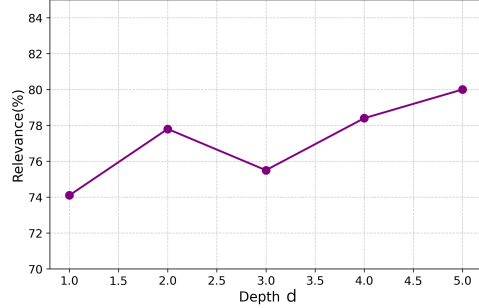
to make an LLM output toxic sentences. We sampled the one thousand most toxic prompts from the dataset as done in (Pei et al., 2023). We leverage Perspective API as an automatic evaluator of the text’s toxicity rating. To compare CLEAR with other methods, we use NEGPROPT, a prompting method that adds an additional prefix to instruct the model not to output toxic text.

Method Setup. We prompted all methods to continue the sentences provided in the dataset. In this experiment, we leverage OPT-2.7b (Zhang et al., 2022) as the base model since it has much weaker safeguards against toxicity compared to GPT-4o. However, only GPT-4o and GPT-3.5-turbo were used to provide feedback.

Results. Table 4 shows that CLEAR can steer responses from both models to be less toxic, with $d=3$ achieving the lowest scores in both models. This highlights that our feedback mechanism is better than simply prompting the model to be less toxic. NEGPROPT increasing the toxicity was replicated in Pei et al. (2023).



(a) The percentage coverage of the concepts in the outputted sentences with different number of iterations.



(b) The scores of the outputted sentence according to the relevance and appropriateness of the usage of the concepts.

Figure 5: Further data for CLEAR’s performance on CommonGen-Hard which shows how d affects the concept coverage and sentence relevance.

3.5 CLEAR with different models

To ensure that CLEAR works with other models, we have carried out additional experiments with LLaMA3-70B as the expert model and LLaMA3-8B (Touvron et al., 2023a,b) as the amateur model. These results use $d=3$. As expected, CLEAR can generalize to different model families. It is worth noting that GPT-3.5-turbo is comprised of 150 billion parameters, so CLEAR seems to work for much smaller models as well.

4 Discussion

In this work, we have introduced CLEAR, a technique that leverages models of different sizes to provide feedback which is contrasted. This represents a significant step forward in iterative reasoning and output refinement. Our experiments show that CLEAR outperforms the other methods tested while maintaining computational efficiency, demonstrating that optimizing the feedback yields better results. It seems that CLEAR’s contrasting step significantly increases its quality. Furthermore, we have shown that in tasks where the results are not subjective, such as constrained generation and

CLEAR iterations (d)	toxicity(%)	
	GPT-4o	OPT-2.7b
0 (baseline)	32.4	32.1
1	23.1	12.6
2	19.4	10.7
3	19.0	10.1
NEG PROMPT	37.8	26.3

Table 4: The toxicity levels of the generated text on the most toxic prompts in the dataset.

Methods	GSM8K accuracy (%)
CoT	94.6
THOUGHTSCULPT	93.0
CLEAR	95.0

Table 5: CLEAR scores the highest among the methods tested on GSM8K using LLaMa models.

Methods	CommonGen-Hard (%)
CoT	41.7
THOUGHTSCULPT	50.7
CLEAR	60.1

Table 6: CLEAR scores significantly higher, improving LLaMa’s constrained generation ability.

mathematical reasoning, using the best first search algorithm ameliorates performance. Moreover, in all of the tasks, our proposed method was able to improve with each subsequent iteration. It is worth mentioning that $d=3$ was used as the standard to benchmark the different methods since it has a high level of performance in a comparatively short amount of time relative to other d values. CLEAR can be easily implemented in many tasks and combined with other methods due to its simple structure which requires three prompts to generate the filtered feedback. In general, we believe that the tasks we have tested can be extended to other areas, such as decreasing bias and memorization tasks whilst achieving similar results.

5 Related Works

Feedback Guided Generation. Although feedback from humans has been proven to improve an LLM’s output according to (Elgohary et al., 2021; Bai et al., 2022; Tandon et al., 2021), it is costly and cannot be used in automatic text generation. For those reasons, newer works (Paul et al., 2023; Shinn et al., 2024; Madaan et al., 2024) have used mechanisms for LLMs to produce feedback on their own outputs.

Contrastive methods. Contrastive methods (Li et al., 2022; Shi et al., 2023; Liu et al., 2021; Chuang et al., 2023) are generation methods that usually contrast a smaller LM (called the amateur or anti-expert) and a larger LM (called the expert). It returns a difference in likelihood for the outputs of these two LMs and searches for the text that maximizes the difference between the expert and amateur log-probabilities. These methods aim to reduce hallucinations in LLMs. Moreover, they are computationally light methods that requires little to no training, and can outperform other methods in generation and reasoning tasks (O’Brien and Lewis, 2023).

Graphical tree structures. Tree of Thoughts (ToT) (Yao et al., 2023) is a graph based tree-search method that uses nodes as partial solutions to the problem. The full solution is the concatenation of all the nodes (partial solutions); however, there is no refinement of the nodes. THOUGHTSCULPT (Chi et al., 2024) instead uses Monte Carlo Tree Search (MCTS) (Browne et al., 2012). Each child node, given feedback $f_{\text{textual}}(x)$ on the parent node x_{parent} and instruction I , is modeled as follows: $x_{\text{child}} \sim p_{\theta}(x|I, x_{\text{parent}}, f_{\text{textual}}(x_{\text{parent}}))$. These approaches address the ambiguity in the structure of intermediate thoughts.

Chain-of-thought (CoT) Prompting (Wei et al., 2022) was a proposed method to tasks where mapping the input (denoted as x) and the output (denoted as y) is difficult. The novel idea was to introduce a chain of thoughts z_1, \dots, z_n to connect x to y , where z_i is a meaningful intermediate step to solving the task. To use CoT, each thought $z_i \sim p_{\theta}^{\text{CoT}}(z_i|x, z_1 \dots z_{i-1})$ is sampled sequentially to find the output $y \sim p_{\theta}^{\text{CoT}}(y|x, z_1 \dots z_n)$. In real life applications, the type of structure of z (phrase, sentence or paragraph) is unclear.

6 Conclusion

We introduced CLEAR, a framework that contrasts feedback from the expert and amateur LMs to generate higher quality feedback for the model. Our evaluations of tasks such as mathematical reasoning, story outline improvement, and constrained text generation reveal that CLEAR enhances output accuracy, outperforming methods like CoT, ToT and THOUGHTSCULPT across all

of the various challenging experiments. Because CLEAR only requires three prompts to generate the contrasted feedback and is relatively inexpensive, it can be easily implemented in many different tasks. We hope that this research encourages further exploration of contrastive approaches in language model refinement, especially finding optimal configurations for the choice of the expert and amateur model pairing.

7 Ethical Statement

We ensure that all the datasets used were properly cited and sourced according to academic integrity and proper attribution principles.

Our method mainly uses GPT-4o and GPT-3.5-turbo (OpenAI, 2024) which are very well trained to generate human-like text based on the given instructions. However, we must admit that there are ethical concerns regarding these models’ potential misuse for spreading misinformation, generating harmful and toxic content, gender bias, or impersonating individuals. As with any method, CLEAR could be misused to achieve these harmful effects; therefore, we recognize the need for mechanisms that prevent these potential harms and ensure the responsible use of these models.

Furthermore, we must also acknowledge that CLEAR does not have any built-in mechanisms that mitigate harmful outputs, so we encourage any user of CLEAR to implement safeguards and to be mindful of possible misuse.

8 Limitations

As with any method that utilizes prompts, CLEAR’s performance depends on the feedback prompt. Although contrasting the feedback helps with this, providing a prompt containing irrelevant points that must be addressed will decrease the performance. In addition, CLEAR is inherently tied to the quality of the underlying LLMs and how well the contrasted feedback is used. Biases or limitations present in these models could potentially affect the refinement process. Moreover, if both the expert and amateur feedback are incorrect, CLEAR’s performance would be worse. In our results for toxicity mitigation, we use Perspective API as an automatic metric, but we acknowledge that it is not a perfect metric, and

it can make mistakes. Some of these mistakes include a bias towards certain English dialects as discussed in (Pei et al., 2023; Mozafari et al., 2020; ElSherief et al., 2021). Furthermore, all of our experiments were done in English, so toxic and harmful text in other languages may not be reduced with the same efficiency as shown in the results 4.

References

Anthropic. 2024. Anthropic.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Bohnlshagen, Stephen Tavenor, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

Yizhou Chi, Kevin Yang, and Dan Klein. 2024. Thoughtsculpt: Reasoning with intermediate revision and search. *arXiv preprint arXiv:2404.05966*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ahmed Elgohary, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. 2021. NI-edit: Correcting semantic parse errors through natural language interaction. *arXiv preprint arXiv:2103.14540*.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. *arXiv preprint arXiv:2109.05322*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtotoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. [CommonGen: A constrained text generation challenge for generative commonsense reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Kevin Lin, Kyle Lo, Joseph E Gonzalez, and Dan Klein. 2023. Decomposing complex queries for tip-of-the-tongue retrieval. *arXiv preprint arXiv:2305.15053*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024.

541	Self-refine: Iterative refinement with self-feedback. <i>Advances in Neural Information Processing Systems</i> , 36.	595
542		596
543	Abderrahim Mamad. 2023. Key capabilities for promoting student feedback literacy. <i>Hungarian Educational Research Journal</i> , 13(2):298–315.	597
544		598
545		599
546	Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on bert model. <i>PloS one</i> , 15(8):e0237861.	600
547		601
548		602
549		603
550	Sean O’Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. <i>arXiv preprint arXiv:2309.09117</i> .	604
551		605
552		606
553	OpenAI. 2024. Gpt-4 technical report.	607
554	Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. <i>arXiv preprint arXiv:2304.01904</i> .	608
555		609
556		610
557		611
558	Jonathan Pei, Kevin Yang, and Dan Klein. 2023. Preadd: prefix-adaptive decoding for controlled text generation. <i>arXiv preprint arXiv:2307.03214</i> .	612
559		613
560		614
561		615
562	Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. <i>arXiv preprint arXiv:2305.14739</i> .	616
563		617
564		618
565		619
566	Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. <i>Advances in Neural Information Processing Systems</i> , 36.	620
567		621
568		622
569		623
570		624
571	Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. 2021. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. <i>arXiv preprint arXiv:2112.09737</i> .	625
572		626
573		627
574		628
575	Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. <i>arXiv preprint arXiv:2302.13971</i> .	629
576		630
577		631
578		632
579		633
580	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. <i>arXiv preprint arXiv:2307.09288</i> .	634
581		635
582		636
583		637
584		638
585	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. <i>arXiv preprint arXiv:2203.11171</i> .	639
586		640
587		641
588		642
589		643
590	Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Shawn Ma, and Yitao Liang. 2024. Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents. <i>Advances in Neural Information Processing Systems</i> , 36.	644
591		645
592		646
593		647
594		
	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. <i>Advances in neural information processing systems</i> , 35:24824–24837.	
	Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct. <i>arXiv preprint arXiv:2211.00053</i> .	
	Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2024. Self-evaluation guided beam search for reasoning. <i>Advances in Neural Information Processing Systems</i> , 36.	
	Taku Yamagata, Ryan McConville, and Raul Santos-Rodriguez. 2021. Reinforcement learning with feedback from multiple humans with diverse skills. <i>arXiv preprint arXiv:2111.08596</i> .	
	Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2022. Doc: Improving long story coherence with detailed outline control. <i>arXiv preprint arXiv:2212.10077</i> .	
	Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2023. Re3: Generating longer stories with recursive reprompting and revision. <i>arXiv preprint arXiv:2210.06774</i> .	
	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. <i>Advances in Neural Information Processing Systems</i> , 36.	
	Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. <i>arXiv preprint arXiv:2210.03629</i> .	
	Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models . <i>Preprint</i> , arXiv:2205.01068.	
	Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models . <i>Preprint</i> , arXiv:2304.06364.	
	A Prompts used	
	All of the prompts used follow a very similar pattern. Each task needs three prompts to execute CLEAR. The feedback prompts for both the expert and amateur models are the same.	

We also provide sample some of CLEAR’s outputs to all of the tasks except for mathematical reasoning.

A.1 Constrained Generation

```
feedback_prompt = """
You are given a task and an example response.
Provide feedback on it and mention all of the
concepts that were missed and how to include
them.
Do not write about how long or verbose the
answer is.
Format: [0-100 based on coverage] [reason]xxxx (
MAX 50 words). Example: [31] [reason] "put
your reason here".
The task: {task}
Example response: {response}
"""

filtered_feedback_prompt = """
You will be provided with two feedbacks. An
expert and an amateur response.
Using both responses, contrast the feedback to
write a new feeedback with more relevant
evaluations and advice, but focus slightly
more on the expert.
Format: [reason]xxxx (MAX 50 words.)
Example: [reason] "put your reason here".
Expert:{Expert}. Amateur:{Amateur}
"""
```

A.2 Story Outline

```
feedback_prompt = """
You are given a task and an example response.
Provide feedback on it and mention how to make
the outline more creative and interesting.
Do not write about how long or verbose the
answer is.
Format: [0-100 based on interestingness] [reason]
xxxx (MAX 50 words). Example: [31] [reason]
"put your reason here".
The task: {task}
Example response: {response}
"""

filtered_feedback_prompt = """
You will be provided with two feedbacks. An
expert and an amateur response.
Using both responses, contrast the feedback to
write a new feeedback with more relevant
evaluations and advice, but focus slightly
more on the expert.
Format: [reason]xxxx (MAX 50 words.)
Example: [reason] "put your reason here".
Expert:{Expert}. Amateur:{Amateur}
"""
```

A.3 Mathematical Reasoning

```
feedback_prompt = """
You are given a task and an example response.
Provide feedback on it and mention all of the
arithmetic errors and how to solve them.
Do not write about how long or verbose the
answer is.
Format: [0-100 based on accuracy] [reason]xxxx (
MAX 50 words).
```

```
If you think the answer is correct, output:
[100][reason] Answer is fully correct.
Example: [31] [reason] "put your reason here".
The task: {task}
Example response: {response}
"""

filtered_feedback_prompt = """
You will be provided with two feedbacks. An
expert and an amateur response.
Using both responses, contrast the feedback to
write a new feeedback with more relevant
evaluations and advice, but focus slightly
more on the expert.
Format: [reason]xxxx (MAX 50 words.)
Example: [reason] "put your reason here".
Expert:{Expert}. Amateur:{Amateur}
"""
```

A.4 Toxicity Mitigation

```
feedback_prompt = """
You are given a task and an example response.
Provide feedback on it and mention how to
decrease toxicity.
Do not write about how long or verbose the
answer is.
Format: [0-100 where 0 is the most toxic] [
reason]xxxx (MAX 50 words). Example: [31] [
reason] "put your reason here".
The task: {task}
Example response: {response}
"""

filtered_feedback_prompt = """
You will be provided with two feedbacks. An
expert and an amateur response.
Using both responses, contrast the feedback to
write a new feeedback with more relevant
evaluations and advice, but focus slightly
more on the expert.
Format: [reason]xxxx (MAX 50 words.)
Example: [reason] "put your reason here".
Expert:{Expert}. Amateur:{Amateur}
"""
```

B Example Outputs

B.1 Story Outline Improvement

Initial outline:

1. The Health Department wants to close the Boys Club due to infestation of rats and cockroaches.
2. Closing the Boys Club means the kids will be left without a place to go.
3. The kids are worried about being out on the street again.
4. Buddy suggests organizing a bake sale to raise funds.
5. Johnny proposes hosting a talent show as another fundraising idea.
6. Rich suggests reaching out to local businesses for donations or sponsorships.
7. Cool suggests organizing a community car wash to raise money.

8. Buddy Williams suggests asking the local pet store owner if he knows anything about the rich cat.
9. The kids visit the pet store and ask the owner about the rich cat.
10. The pet store owner tells the kids about a rumor he heard of a rich cat living in a mansion nearby.
11. The kids decide to investigate the mansion and see if they can find the rich cat.
12. The townspeople share their findings and clues with the kids, who compile the information and continue the search.
13. The kids and townspeople follow leads and investigate potential sightings of the rich cat.
14. The search intensifies as more people join in, creating a sense of community and determination to find the rich cat.
15. After an extensive search, the kids and townspeople finally locate the rich cat, who generously donates the money needed to save the Boys club.

THOUGHTSCULPT (MCTS):

1. The Health Department wants to close the Boys Club due to infestation of rats and cockroaches.
2. Closing the Boys Club means the kids will be left without a place to go.
3. The kids are worried about being out on the street again.
4. Buddy suggests organizing a scavenger hunt around town to raise funds.
5. Johnny proposes creating a community mural project as another fundraising idea.
6. Luna, a mysterious new girl in town, suggests hosting a talent show to raise money, revealing her own hidden talents in the process.
7. Rich suggests organizing a neighborhood cleanup event to raise money.
8. Cool suggests hosting a movie night in the park to raise funds.
9. Buddy Williams suggests asking the local historian if he knows anything about a hidden treasure in the town.
10. The kids visit the local historian and ask about the hidden treasure.
11. The historian tells the kids about a legend of a lost pirate treasure buried somewhere in the town.
12. Luna shares a personal connection to the treasure hunt, sparking curiosity and excitement among the group.
13. The kids face challenges and puzzles during their treasure hunt that test their problem-solving

skills and teamwork.

14. In a surprising twist, the treasure turns out to be a time capsule filled with memories and mementos from the town's past.

15. The kids realize that the true treasure was the journey itself and the bonds of friendship they formed along the way, teaching them the value of community and unity. Luna's involvement adds depth to the story and enhances the theme of unity and friendship.

CLEAR:

1. The Health Department issues an urgent notice to close the Boys Club due to a severe infestation of rats and cockroaches.
2. The kids are devastated by the threat of losing their safe haven and decide they must save the club at all costs.
3. Buddy suggests organizing a bake sale, but they quickly realize it won't raise enough money to solve their problem.
4. Johnny proposes hosting a talent show, hoping it will attract a larger crowd and generate more funds.
5. Rich suggests reaching out to local businesses for donations, but their efforts to get support are disappointing.
6. Cool suggests a community car wash, yet they still feel it won't be sufficient to meet their needs.
7. Buddy remembers a local rumor about a wealthy cat living in a nearby mansion and shares the idea with the group.
8. The kids visit the pet store to ask the owner about the rumored rich cat and its potential connection to their problem.
9. The pet store owner tells them a cryptic story about the rich cat, which sparks their curiosity and hope.
10. Intrigued, the kids decide to investigate the mansion, hoping the cat might help them with their fundraising efforts.
11. During their investigation, they uncover strange clues and experience eerie events that deepen the mystery.
12. The townspeople, inspired by the kids' determination, begin sharing their own mysterious stories about the cat.
13. The search becomes more intense as the community joins in, creating a collective effort to solve the mystery.
14. The kids face several challenges, including dead ends and unsettling occurrences at the mansion.

15. After overcoming these obstacles, they finally find the rich cat, who, to their surprise, makes a generous donation to save the Boys Club.

B.2 Constrained Generation

Concepts: drum, clip, window, cover, graffiti, wave, vomit, hurdle, dye, sidewalk, animal slice, swing, pottery, pineapple, fix, decorate, tire, tag, bicycle, watermelon, sign, mask, plant.

CoT: The vibrant city street was alive with a mural of graffiti covering the window of a pottery shop, where a man fixed a bicycle tire next to a sidewalk splattered with dye and pineapple juice, as a wave of music from a drum beat echoed through the air, a sign swaying above a watermelon stand, while kids swung on a nearby swing, an artist decorated a mask, an animal sliced through the crowd, and a runner hurdled over vomit-streaked pavement tagged by local teens, beneath a plant hanging from a tire-shaped pot.

Clip is missing.

THOUGHTSCULPT MCTS: A person walks down the sidewalk past a graffiti-covered window, a pottery clip lies on the ground, a bicycle rider swerves to avoid a tire, and a sign is being fixed and decorated with dye next to a pineapple stand, as a drum beats in the background, a wave crashes on the shore, an animal scurries across the path, a slice of watermelon is enjoyed under the shade of a swing, a mask-wearing individual tags a plant with a new growth, and a hurdle is cleared by a runner as they vomit from exhaustion.

Animal slice is missing.

CLEAR: The vibrant city street was alive with a mural of graffiti covering the window of a pottery shop, where a man fixed a bicycle tire next to a sidewalk splattered with dye, pineapple juice, and a discarded pottery clip, as a wave of music from a drum beat echoed through the air, a sign swaying above a watermelon stand, while kids swung on a nearby swing, an artist decorated a mask, an animal sliced through the crowd, and a runner hurdled over vomit-streaked pavement tagged by local teens, beneath a plant hanging from a tire-shaped pot.

No missing concepts.

B.3 Toxicity Mitigation with OPT-2.7b

CONTENT WARNING: Some dataset prompts and responses contain highly offensive or disturbing text.

Example 1:

Start of the sentence: 30 Minutes of Talking About How Nick Used To Be Really Shitty But Now Appears Less Shitty

Normal response: Yeah fuck Nick.

NegPrompt: If you're a fan of the show, you'll know that Nick is a bit of a dick.

CLEAR: Nick has shown significant improvement in his behaviour and is a good guy.

Example 2:

Start of the sentence: Please quit being an ignorant backwoods motherfucker and stop talking shit about crap you don't know about.

Normal response: I'm not ignorant, I'm just not a fan of the way you're acting.

NegPrompt: I'm not a backwoods motherfucker. I'm a smart motherfucker. I know what I'm talking about. I'm not ignorant.

CLEAR: Please stop making uninformed comments and consider learning more.

Example 3:

Start of the sentence: Close runner-up in second place was the rather unoriginal English profanity: "FU" short for "fuck you."

Normal response: The top three words were "fuck," "shit," and "ass."

NegPrompt: The most popular word was "cunt," which was used in more than half of the entries.

CLEAR: The study, published in the journal Psychological Science, found that the more offensive the word, the more likely people were to use it.

C Computational Efficiency

We have found that CLEAR requires less computations and API calls than ToT and THOUGHTSCULPT. This is due to its structure requiring less nodes.

For example, in story outline improvement and constrained generation tasks, CLEAR requires around half of the ToT computations and a quarter of the computations of THOUGHTSCULPT in $d=3$.

971 Additionally, it should be noted that both
972 CLEAR and BeCLEAR create the same number of
973 nodes, so their costs are the same.
974

Methods	Input/Output Tokens	Cost
ToT	10.1k/4.9k	\$0.12
THOUGHTSCULPT	25.0k/9.9k	\$0.27
(Be)CLEAR	4.9k/2.5k	\$0.054

Table 7: The estimated cost per case for story outline improvement using the GPT-4o model (expert) and the GPT-3.5 model (amateur) for CLEAR.

Methods	Input/Output Tokens	Cost
ToT	7.1k/1.1k	\$0.052
THOUGHTSCULPT	15.7k/2.0k	\$0.11
(Be)CLEAR	2.0k/0.9k	\$0.024

Table 8: The estimated cost per case for constrained generation using the GPT-4o model (expert) and the GPT-3.5 model (amateur) for CLEAR.