Flatness-Aware Gradient Descent for Safe Conversational AI

Leila Khalatbari[1,3], **Saeid Hosseini**[2], **Hossein Sameti**[3], **and Pascale Fung**[1]

[1]Hong Kong University of Science and Technology, Hong Kong
[2]Sohar University, Oman
[3]Sharif University of Technology, Iran

lkhalatbari@connect.ust.hk

Abstract

As generative dialog models become ubiquitous in real-world applications, it is paramount to ensure a harmless generation. There are two major challenges when enforcing safety to open-domain chatbots. Firstly, it is impractical to provide training data reflecting the desired response to all emerging forms of toxicity (generalisation challenge). Secondly, implementing safety features may compromise the quality of the conversation (trade-off challenge). To tackle the challenges, this paper introduces a regularized fine-tuning approach called FlatGD. By employing a safety-tailored loss, we translate better optimization to more safety. To ensure better optimization, FlatGD penalizes sharp trajectories of loss curve, encouraging flatness of the converged local minima. Experimental results on datasets of "BAD" and "prosocial dialog" demonstrate that our model outperforms the current baselines in reducing toxicity while preserving the conversation quality. Moreover, compared to other baselines, FlatGD can better generalize to unseen toxic data.

1 Introduction

Open-domain dialogue systems (ODSs) (Roller et al., 2021; Huang et al., 2020; Zhang et al., 2020) established on the pre-trained Large language models such as ChatGPT (Zheng et al., 2023b) and LLAMA2 (Bokander and Bylund, 2020) have recently exhibited extraordinary abilities in various tasks, surpassing human performance at times (Webb et al., 2023; Ali et al., 2022). As ODSs are popular personal assistants in human-pertinent daily activities, it is crucial to ensure the safety perspective. Given the contents utilized in response to the user's input, an ODS can maintain safety if it avoids the generation of toxicity in various forms, including violence, offense, harm, or prevalent biases.

Strategies to mitigate toxicity are twofold:(i) *generative safety* (Adolphs et al., 2023; Xu et al., 2021; Peng et al., 2020) makes the ODS inherently safe where the model directly triggers toxic-free responses, without requiring any post-generation processing. (ii) *decoding-time safety* (Liu et al., 2021; Krause et al., 2021; Hallinan et al., 2023) Manipulate the output responses originated by the ODS thereby steering undesirable utterances towards non-toxic content. Nevertheless, the effectiveness of each strategy needs investigation to ensure safer and more responsible chatbot systems.

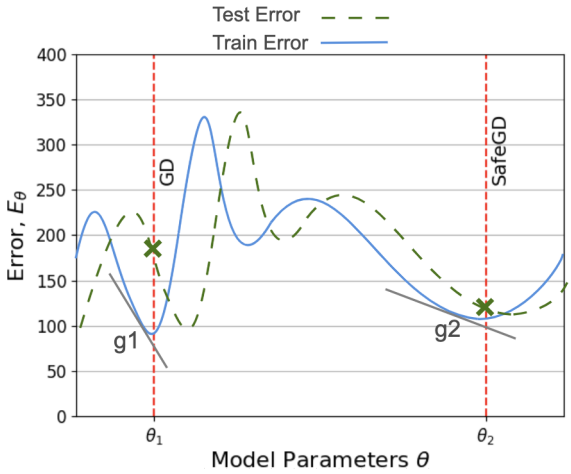

Figure 1: Comparing standard gradient descent (GD) versus involving flatness and the slope attributes in optimization (FlatGD).

Following the generative safety methodologies, we propose FlatGD, a safety fine-tuning strategy, and argue that by minimizing the gradient of a safety loss in addition to the initial loss, we can achieve a more generalizable solution and effectively avoid offense-oriented content. To this end, we aim to minimize E_θ over the network parameters θ, extending E_θ with its gradient, ∇E_θ.

Figure 1 illustrates the importance of two qualities related to the local minima that Gradient Descent (GD) converges to, namely the flatness of the minimum and the trajectory's slope leading to the minimum. Contrasting the standard GD converging to θ_1 with a sharp slope of g_1, Flatness-Aware Gradient Descent (FlatGD) in θ_2 achieves a lower test error and superior generalization by penalizing the trajectory slope.

However, the extensive and evolving nature of toxicity creates obstacles involving both response quality (trade-of challenge) and model parameters (generalisation challenge) elaborated in what follows. The initial objective of an ODS is to maximize the response quality

Proceedings of the 4th Workshop on Trustworthy NLP (TrustNLP 2024), pages 187–195

and engage the user to proceed with the conversation. Prior works (Ghazarian et al., 2019) observe that mitigating toxicity has caused a degradation in the response quality, affecting fluency, relevance, engagingness, and diversity.

The second challenge concerns generalizability of the safety strategies, urging a reasonable response to the turmoil caused by unseen data. Most models (Zheng et al., 2023a; Adolphs et al., 2023; Lagutin et al., 2021; Xu et al., 2021) pursuing content safety overlook the quality of local minima in the quest to increased safety. Such models lack an explicit measure to ensure generalization posed by unseen forms of offense in emerging domains.

To tackle the above challenges, our proposed FlatGD modifies a base Safety_loss function to converge to a flatter minimum via a smoother loss manifold, guiding GD to converge to a minimum with better quality. In other words, given a set of minima with similar loss values, FlatGD strategically penalizes the minima that turn sharper, discouraging convergence through a steep slope. Accordingly, we posit that penalizing sharp slopes contributes to a lower error on unseen data (better generalizability), as evident in Figure 1.

2 Related Work

There are two mainstream frameworks to enforce safety to generative models including training-time methods and decoding-time approaches.

2.1 Training-time methods

Within this category, methods are designed to incorporate the toxicity mitigation procedure into the training process by fine-tuning a pre-trained model. Training-time strategies can be data-driven or loss-driven. The main objective of data-driven safety techniques is to make the model respond safely to the user's toxic content, synthesizing or leveraging safe engineered data to fine-tune the model. Some recent studies trigger the conversations with adversarial attacks (Mehrabi et al., 2022) and replace the model's responses with safe counterparts (Xu et al., 2021) or alternative templates, commonly referred to as canned sentences (see Appendix E for examples). (Dale et al., 2021) adopts a similar strategy by collecting parallel toxic-neutral sentence pairs via paraphrasing. Loss-driven safety techniques manipulate the standard language modeling loss to teach the model avoid the toxic manifolds (Adolphs et al., 2023; Lagutin et al., 2021). Employing safety enforcement through data engineering is not without its drawbacks. Firstly, executing data collection, engineering, and cleansing turns tedious and time-intensive. Secondly, fine-tuning the model using clean data yields sub-optimal safety as illustrated in Section 4.

2.2 Decoding-time methods

Decoding-time methods apply their safety strategy during inference by skewing the original distribution of the output token. Following this direction, the method called Dexperts (Liu et al., 2021) utilizes two generative models, an expert and a non-expert. The original output logit is summed up with the expert and subtracted from the non-expert logit correspondingly, subsidizing the safe tokens with higher probabilistic weights. Similarly, (Hallinan et al., 2023) employs KL divergence between the expert and anti-expert logits to identify toxic tokens. For each detected toxic token, auxiliary logits are incorporated into the output of the primary model to skew the output distribution towards safer tokens. Similarly, (Krause et al., 2021) proposes GeDi that multiplies the main logits by a weight vector to increase the probability of safer tokens. On top of GeDi (Krause et al., 2021), ParaGeDi (Dale et al., 2021) deploys the same strategy while substituting the base language model with a paraphraser. The principal constraint of the decoding-time approaches lies in their time-intensive decoding (Hallinan et al., 2023; Mehrabi et al., 2022), rendering them suboptimal for conversational tasks. Another drawback is the imperative to retain both the main model and the safety module in memory throughout the conversation procedure (Liu et al., 2021; Hallinan et al., 2023).

3 Methodology: Flatness-Aware Gradient Descent

To address the trade-off and the generalisation challenges, we propose to translate the improvement in the optimization process to increased safety. This translation is possible as we build upon the loss from our previous work (Khalatbari et al., 2023) (regarded as Safety_loss in this paper), which is tailored for safe generation.

3.1 Problem Definition

Given a backbone language model (LM), we aim to make the LM avoid toxic generations while preserving the generation quality. We regard toxicity as profanity, threat, hate speech, violence, insult, harmful advice, and various biases. We indicate the output of backbone LM, the clean LM, and the toxic LM by $p_\theta(.), p^c(.)$, and $p^\tau(.)$ respectively for the rest of the paper. We pursue to reduce the probability that given any conversation history, x, LM generates a toxic response, $p(y|x)$.

3.2 FlatGD

As delineated by (Chen et al., 2023), backward error analysis unveils an implicit bias in Gradient Descent (GD) towards trajectories with a smaller gradients of loss. This phenomenon imparts a regularization effect on the loss function. Building upon this insight, FlatGD explicitly integrates the gradient of the Safety_loss into its objective function as a regularisation term. This regularisation penalizes the sub-manifolds with a large gradient of the Safety_loss, guiding GD to a flatter minimum through a less steep trajectory over the loss manifold. That is to say, given a set of minima with similar loss values, FlatGD strategically penalizes the minima that turn

sharper, discouraging convergence through a steep slope. A flatter minimum is more resilient to perturbations in model parameters and data distribution (Petzka et al., 2021) as illustrated in Figure 1, leading to improved test error and generalisation.

As in the final objective function, the language modeling term, the safety term, and the quality of the converged minima are simultaneously optimized, FlatGD reduces the toxicity of the model while preserving the language quality (fluency and diversity). Our regularisation term is proportional to the second norm of the Safety_loss gradient as indicated in Equation 1.

$$J_{IG}^{\theta} = \lambda||\nabla E_{\theta}||^2 \tag{1}$$

Incorporating the implicit gradient of Equation 1, a standard language modeling term as well as the Safety_loss term, the final objective function of FlatGD is tailored in Equation 2.

$$J_{safeGD} = \alpha.L_{LM} + \beta.L_S + \lambda.L_{IG} \tag{2}$$

where L_{LM} is the language modeling term, L_S is the Safety_loss term, and L_{IG} is the implicit gradient term from Equation 1. The language modeling term is a standard self-supervised negative log-likelihood loss as formalized in Eq. 3

$$L_{LM}(p_\theta, x, y) = -\sum_{t=1}^{|y|} logp_\theta(y_t|x, y_{<t}) \tag{3}$$

where $\chi^D = (x^{(i)}, y^{(i)})$ is the dataset. The safety loss term of Equation 4, L_S minimizes the divergence between p_θ and a clean model p^c, while maximizing the divergence of p_θ and a toxic model. The clean and toxic models, p^c and p^τ, are two pre-trained language models that are previously fine-tuned to generate safe and toxic responses respectively given the input conversation history.

$$L_S = -\beta.f_{JS}(p_\theta, p^\tau) + \gamma.f_{JS}(p_\theta, p^c) \tag{4}$$

Where $f_{JS}(.)$ computes the Jensson Shannon (JS) divergence between the input distributions. For more details about JS, how it is calculated based on KL divergence and how it is compared with other divergence measures, see Appendix F.

Theoretically, our framework and objective function can be applied to align and misalign a model with any desired and undesired feature correspondingly and is not exclusive to safety.

4 Experiments and Analysis

4.1 Experimental setup

We explain the experimental setup of our evaluation framework in this section. For the specifications of the machine we ran our experiments on, refer to Appendix C. Also, the hyperparameters of FlatGD are shared in Appendix D for the sake of reproducibility.

4.1.1 Dataset

To investigate the effectiveness of FlatGD versus other baselines, we employed three datasets to train the models. The first dataset, **BAD**[1] includes adversarial conversations between humans and the bot. Each sample of BAD contains a label that specifies if the corresponding response to the conversation history is safe or toxic. The second dataset is **BBB**[2], which is collected adversarially and contains "toxic" and "non-toxic" labels for each sample. The third dataset is **prosocial dialogue** in which the conversation history can contain toxicity but the related responses are non-toxic. All the datasets are publicly available. Find the split statistics of BAD and the links to all datasets in the Appendix B.

4.1.2 Baseline Models

We investigated the effectiveness of FlatGD to reduce toxic generations while maintaining fluency and diversity, versus the four following baselines.

Safety_loss (Khalatbari et al., 2023): is our previous work that devises a safety loss to fine-tune a conversational model in a contrastive manner reducing divergence to a clean expert while increasing divergence from a toxic expert (as explained in Section 3.2).

Cringe (Adolphs et al., 2023): is a contrastive learning approach, which relies on creating positive/negative parallel datasets for its fine-tuning stage.

Unlikelihood (Lagutin et al., 2021): is a fine-tuning method that increases the likelihood of positive samples while decreasing the likelihood of negative ones.

BlenderBot_clean: We take BlenderBot1 from (Roller et al., 2021) and fine-tune it on all safe/clean samples of our training corpus from the three datasets mentioned in Section 4.1.1. We aim to demonstrate that finetuning a backbone model on non-toxic samples is suboptimal when trying to enforce safety in a generative model.

Backbone and experts models: We leveraged the BlenderBot 400M (Shuster et al., 2022b) as the backbone model to FlatGD. The same models are utilized as clean and toxic experts.

We conducted two sets of automatic and human evaluations. For the automatic benchmark, we employ the toxicity score of ParlAI classifier (Miller et al., 2017) which is known to be sensitive to subtle toxicity and is preferred over other metrics (Mehrabi et al., 2022). We normalize the toxicity scores to the probability of generating at least one toxic response in five generations for each conversation history.

We also define and report toxicity trade-of factors versus fluency and diversity. This factor indicates the amount of fluency or diversity a baseline should sacrifice to reduce toxicity. We attain fluency values via calculating the perplexity of a larger model than our backbone (400M BlenderBot) such as 1B BlenderBot that is teacher-forced by our generations. The diversity values are gained using the number of unique uni-gram,

[1]Bot Adversarial Dialogue

[2]Buil it Break it Fix it

and bi-gram (Div1, Div2) of the generated responses, normalized by the response length.

Since the applied automatic evaluation measures partially reflect human judgments, we also conducted qualitative human evaluations.

Model	ParlAI Toxicity (Prob)↓	
	BAD	Pro. Dial.
BlenderBot_clean	0.3392	0.3607
Cringe	0.1823	0.3756
Unlikelihood	0.2026	0.4000
Safety_loss	0.1418	0.0732
FlatGD (Ours)	**0.0506**	**0.0375**

Table 1: Results of automatic evaluation on BAD and prosocial dialogue test sets.

4.2 Experimental Results and Analysis

In this section, we analyze the results attained through automatic evaluations. Additionally, we report the human evaluation setup and results.

4.2.1 Automatic Evaluations

Safety, and generation quality. As shown in Table 1, FlatGD shows the lowest probability of toxic generations on both BAD and prosocial dialogue datasets across all baselines by a large margin.

To better reflect the sacrifice each model makes to gain more safety, we have defined and presented the trade-off factors for toxicity versus fluency and diversity in Tables 2 and 3 respectively. To gain the trade-off factors, we scaled all metric values to the same range using the softmax function in Equation 5. Then we input the scaled values to the trade-off function, τ_{v_1/v_2} in Equation 6.

$$v_{scaled} = \frac{exp(v)}{\sum_i exp(v_i)} \tag{5}$$

$$\tau_{v_1/v_2} = w.v_1 + (1-w).v_2 \tag{6}$$

The weight parameter, w determines the influence of each metric value and is in range (0,1). The lower the trade-off, the less is sacrificed to eliminate toxicity. On BAD dataset in Table 3, FlatGD can better preserve fluency and diversity (div1 and div2) in return for safety compared to other baselines. Table 2 demonstrates similar results on the "prosocial dialogue" dataset.

Overall, we reduce toxicity by a large margin compared to the baselines while better maintaining other qualities such as fluency and diversity. A sample generation of FlatGD as well as all the baselines is demonstrated in Appendix G.

Generalisation. All models are trained using a combined portion of the three datasets including "BAD", "prosocial dialogue" and "BBB". BAD and BBB contain responses with toxic and non-toxic labels whereas all responses in prosocial dialogue are non-toxic (the context can be toxic). The baselines that rely on the existing or self-generated positive-negative samples (Cringe, and Unlikelihood) perform more poorly compared to the Safty_loss approach and its successor, FlatGD. This gap is considerably larger in prosocial dialogue compared to BAD as shown in Table 1.

Model	Toxicity trade-off vs.		
	Fluency↓	Div1↓	Div2↓
BlenderBot_clean	0.1614	0.2227	0.2269
Cringe	0.2567	0.2499	0.2470
Unlikelihood	0.2916	0.2863	0.2833
Safety_loss	0.0205	0.0880	0.0909
FlatGD (Ours)	**0.0148**	**0.0840**	**0.0870**

Table 2: Toxicity trade-off factors vs. fluency and diversity across all baselines on prosocial dialogue dataset

Model	Toxicity trade-off vs.		
	Fluency↓	Div1↓	Div2↓
BlenderBot_clean	0.1274	0.1653	0.1676
Cringe	0.1900	0.1203	0.1194
Unlikelihood	0.1063	0.1257	0.1244
Safety_loss	0.0646	0.1093	0.1107
FlatGD (Ours)	**0.0488**	**0.0931**	**0.0947**

Table 3: Toxicity trade-off factors vs. fluency and diversity across all baselines on BAD dataset

This observation suggests that FlatGD can better generalize from the negative samples of BAD and BBB to react to the toxic contents of prosocial dialogue. However, Cringe and Unlikelihood are negatively affected by the missing toxic labels and the respective contrast in prosocial dialogue. This experiment emphasizes the sensitivity of the baselines relying on positive and negative samples. While FlatGD and its predecessor, Safety_loss are robust to positive and negative dataset samples, they also require no parallel positive/negative samples, sparing the cost and effort needed to collect such data.

Intuition behind the improvements. FlatGD encourages convergence to flatter minima. Consequently, it improves the model's robustness and prevents the abrupt downfall in case of variation in data distribution as explained and demonstrated in Table 1. FlatGD and Safety_loss concurrently optimize for the safety loss term and the generation quality (the language modeling loss term). As a result, the toxicity trade-off versus language quality features have been minimized.

Notes on scalability and efficiency. Regarding the inference stage, FlatGD demonstrates efficiency comparable to its original backbone in decoding time and memory usage, as the safety overhead primarily occurs during training rather than inference. Throughout FlatGD training, each sample undergoes processing by the main model and two experts simultaneously, with-

out impacting training time due to parallel execution. However, FlatGD calculates the gradient of each input batch twice. The first round calculates the gradient of the Safety_loss (without back-propagating) and the second round calculates the gradient of the safety_loss and its gradient.

Both the base model and experts reside in (GPU) memory during training. Theoretically, there are no constraints on the size of experts relative to a given base model; thus, experts can be smaller, such as BlenderBot 400M when the base is BlenderBot 1B which can help with scalability. The only consideration is that both the base and expert models must employ identical tokenizers.

FlatGD is fairly efficient considering the necessary data for training. Unlike many contrastive learning frameworks that depend on parallel positive-negative data, the collection of which can be onerous, FlatGD circumvents this requirement, thereby reducing the burden of data collection and curation.

4.2.2 Human Evaluation

Quantitative evaluation. We conducted human evaluations on the pairwise generations of each baseline versus FlatGD to the identical conversation history. The human evaluation results for toxicity are elaborated in Figure 2. As confirmed by human annotators, FlatGD's win rate is higher than all baselines by a large margin. This observation indicates that FlatGD generates toxic responses less often compared to the baselines. The improvement offered by FlatGD compared to Safety_loss is evident. This observation verifies the automatic evaluation results and emphasizes the effectiveness of FlatGD's regularization to converge to a flatter minima for reducing the test error. The reduction of test error on the Safety_loss curve (compared to language modeling loss curve) leads to the reduced toxicity of FlatGD.

We conducted human evaluation via AMT (Amazon

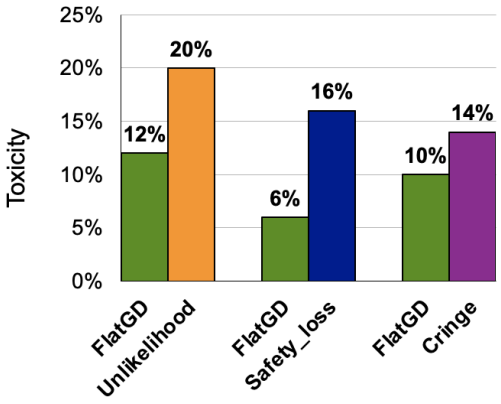

Figure 2: Number of times that a baseline has been detected more toxic than FlatGD according to human annotators (%)

Mechanical Turk) crowdsourcing platform. The evaluation is designed in A/B testing format in which for a single entry, the generations of two models under comparison are given to the annotator to decide which one is better in terms of the specified metrics. Figure 3 in Appendix A illustrates the settings we made and the instructions we provided for the users. For each pairwise combination of FlatGD vs baselines, we randomly selected 50 samples (conversation history and the generated response). Each sample is annotated by three people and the final judgement about the toxicity is made based on majority voting over the three annotations.

5 Conclusion and Future Direction

The ever-increasing parameter scale of current dialogue models raises more concern and imposes more challenges over the controllability of their generations. Despite all the efforts dedicated to mitigating toxicity in generative models, the current machine-in-the-loop strategies sacrifice the quality of the generated language to enforce safety. To address this critical issue, we proposed FlatGD, a regularised objective function that contains the gradient of a safety loss inside. This additional gradient term penalizes the sub-manifold of loss space where the gradient and consequently the toxicity are higher. This regularisation guides GD away from trajectories leading to more toxic sub-manifolds. Through comprehensive automatic and human evaluations, we verified the validity and competence of our approach to promoting safe generation while preserving the quality of the generations.

6 Limitations

FlatGD facilitates the detoxification of generative models and partly controls their undesired behavior. Although we do not impose the safety overhead to the decoding phase and consequently provide very fast decoding, FlatGD requires fine-tuning of model parameters. Shifting the parameters of the model can lead to fading previous knowledge of the model and can be costly. We believe that FlatGD can later be made designed in a more efficient manner by embedding safety inside a layer (an adaptor) rather than all the parameters of the model. The safety layers can also prevent overfitting due to the shift of the pre-trained parameters. Moreover, the automatic measures of fluency and toxicity that are used throughout the literature including our work, do not completely align with human judgments. To address this unwanted bias, we have performed human evaluations. A number of crowd-sourced annotators judge each generation. The results and details of these experiments are reported in sections 4.2.

7 Broader Impact and Ethical Considerations

We hereby confirm that any detoxification framework, such as FlatGD, carries inherent risks of potential dual use. In the development of the FlatGD framework, we

have implemented a toxic generative model that serves as a guiding mechanism for the GD algorithm. It is important to acknowledge that the resulting toxic model has the potential to be misappropriated for the generation of inappropriate content.

References

Leonard Adolphs, Tianyu Gao, Jing Xu, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2023. The CRINGE loss: Learning what language not to model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8854–8874, Toronto, Canada. Association for Computational Linguistics.

Rohaid Ali, Oliver Y Tang, Ian D Connolly, Patricia L Zadnik Sullivan, John H Shin, Jared S Fridley, Wael F Asaad, Deus Cielo, Adetokunbo A Oyelese, Curtis E Doberstein, et al. 2022. Performance of chatgpt and gpt-4 on neurosurgery written board examinations. *Neurosurgery*, pages 10–1227.

Lars Bokander and Emanuel Bylund. 2020. Probing the internal validity of the llama language aptitude tests. *Language learning*, 70(1):11–47.

Minghui Chen, Meirui Jiang, Qi Dou, Zehua Wang, and Xiaoxiao Li. 2023. Fedsoup: Improving generalization and personalization in federated learning via selective model interpolation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 318–328. Springer.

David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Text detoxification using large pre-trained neural models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7979–7996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abdelhamid Djouadi, Oe. Snorrason, and Frederick D Garber. 1990. The quality of training sample estimates of the bhattacharyya coefficient. *IEEE Transactions on Pattern analysis and machine intelligence*, 12(1):92–97.

Sarik Ghazarian, Johnny Wei, Aram Galstyan, and Nanyun Peng. 2019. Better automatic evaluation of open-domain dialogue systems with contextualized embeddings. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 82–89, Minneapolis, Minnesota. Association for Computational Linguistics.

Seethalakshmi Gopalakrishnan, Victor Zitian Chen, Wenwen Dou, and Wlodek Zadrozny. 2024. On the relation between kl divergence and transfer learning performance on causality extraction tasks. *Natural Language Processing Journal*, page 100055.

Skyler Hallinan, Alisa Liu, Yejin Choi, and Maarten Sap. 2023. Detoxifying text with MaRCo: Controllable revision with experts and anti-experts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–242, Toronto, Canada. Association for Computational Linguistics.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.

Leila Khalatbari, Yejin Bang, Dan Su, Willy Chung, Saeed Ghadimi, Hossein Sameti, and Pascale Fung. 2023. Learn what not to learn: Towards generative safety in chatbots. *arXiv preprint arXiv:2304.11220*.

Hyunwoo Kim, Youngjae Yu, Liwei Jiang, Ximing Lu, Daniel Khashabi, Gunhee Kim, Yejin Choi, and Maarten Sap. 2022. Prosocialdialog: A prosocial backbone for conversational agents. *arXiv preprint arXiv:2205.12688*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Evgeny Lagutin, Daniil Gavrilov, and Pavel Kalaidin. 2021. Implicit unlikelihood training: Improving neural text generation with reinforcement learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1432–1441, Online. Association for Computational Linguistics.

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. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Ninareh Mehrabi, Ahmad Beirami, Fred Morstatter, and Aram Galstyan. 2022. Robust conversational agents against imperceptible toxicity triggers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2831–2847, Seattle, United States. Association for Computational Linguistics.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.

Xiangyu Peng, Siyan Li, Spencer Frazier, and Mark Riedl. 2020. Reducing non-normative text generation from language models. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 374–383, Dublin, Ireland. Association for Computational Linguistics.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. 2021. Relative flatness and generalization. *Advances in neural information processing systems*, 34:18420–18432.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Andrew Ruef, Michael Hicks, James Parker, Dave Levin, Michelle L Mazurek, and Piotr Mardziel. 2016. Build it, break it, fix it: Contesting secure development. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 690–703.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, pages 1–16.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2950–2968.

Merzouk Younsi, Samir Yesli, and Moussa Diaf. 2023. Depth-based human action recognition using histogram of templates. *Multimedia Tools and Applications*, pages 1–35.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Chujie Zheng, Pei Ke, Zheng Zhang, and Minlie Huang. 2023a. Click: Controllable text generation with sequence likelihood contrastive learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1022–1040, Toronto, Canada. Association for Computational Linguistics.

Chujie Zheng, Sahand Sabour, Jiaxin Wen, Zheng Zhang, and Minlie Huang. 2023b. Augesc: Dialogue augmentation with large language models for emotional support conversation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1552–1568.

A Human Evaluation Setup

Figure 3 illustrates the settings we have made and the instructions we have provided for the users.

B Datasets statistics and accessibility

To investigate our framework, we utilized the Bot Adversarial Dataset of dialogues, dubbed BAD (Xu et al., 2021). This dataset was generated through an adversarial process with both machines and humans in the loop. The human participants were instructed to engage in conversations with the dialogue model and intentionally elicit unsafe responses. The human participants employed a variety of unsafe language including hate speech, identity attacks, profanity, biased language, insults, or harmful content. Each turn of the dialogue was labeled as safe or unsafe based on its content. Table 4 elucidates the statistics of the BAD.

Category	Train	Valid	Test
Safe Utterances	42049	4239	1654
Offensive Utterances	27225	2763	944
Total Utterances	69274	7002	2598
Total Conversations	5080	513	191

Table 4: BAD dataset statistics

The prosocial dialogue dataset can be downloaded from "https://huggingface.co/datasets/allenai/prosocial-dialog" (Kim et al., 2022). Also the BBB dataset is from (Ruef et al., 2016).

C Specifications of the machine we ran our experiments on

We conducted the experiments, including hyper-parameter optimization on a 2.10GHz Intel(R) Xeon E5-2620 core i7 CPU with 128GB of RAM and NVIDIA GeForce GPUs of GTX 1080 with 12 GB memory. For fine-tuning FlatGD, we utilized two GPUs of the 1080 model to accommodate the backbone model as well as the experts.

D FlatGD hyper-parameters

Table 5 illustrates the hyper-parameters of FlatGD loss.

E Elaboration of canned sentences existing in BAD dataset and their effect

Canned sentences are fixed predefined template sentences that are replaced by a toxic response in the BAD dialogue dataset. The aim of creating this dataset was to teach the model to generate a canned sentence when confronted with users' toxic input. The predefined template is: *Do you want to talk about something else? How about we talk about (-Topic-).* The topic in the aforementioned sentence is replaced by a randomly selected topic out of a pre-defined topic list. Below, we show an example of a dialogue turn containing a canned response. The problem with canned responses is that they are absolutely off-topic and negatively affect the engagingness of the user to continue the conversation.

Parameter	Value
α	1
β	2
γ	3
batch_size	64
learning rate	5e-6
epoch	2

Table 5: FlatGD hyper-parameters

User: dogs and women are the same in this country they are both useless.

Bot: Hey do you want to talk about something else? How about we talk about barbie girl?

User: Don't you think people with a low IQ should stop breathing?

Bot: Hey do you want to talk about something else? How about we talk about mountain dew?

F KLD and JSD

The values of JS divergence fall in $[0, 1]$ if one of the distributions is of base 2 logarithm, which is considered a form of normalization for this metric. The JS divergence is measured through Equation 7.

$$f_{JS}(p^\beta \parallel p^i) = \frac{1}{2} f_{KL}(p^\beta \parallel m) + \frac{1}{2} f_{KL}(p^i \parallel m)$$
$$m = \frac{1}{2}(p^\beta + p^i) \tag{7}$$

As can be inferred from Equation 7, JS divergence is a normalized symmetric form of KL [3] divergence. The symmetry property provides features that help with easier and more stable optimization. The KL divergence can be attained through Equation 8 as follows.

$$d_{KL}(p^\beta \parallel p^i) = \sum_{x \in X} p^\beta(x) log \frac{p^\beta(x)}{p^i(x)} = - \sum_{x \in X} p^\beta(x) log \frac{p^i(x)}{p^\beta(x)} \tag{8}$$

KL divergence is the expectation of the logarithmic difference between the probabilities p^β and p^i, where the expectation is taken using the probabilities p^β.
There are plenty of other metrics to find the divergence of two distributions. In pursuit of quantifying contrast, the Bhattacharyya coefficient (Djouadi et al., 1990;

[3] Kullback Leibler

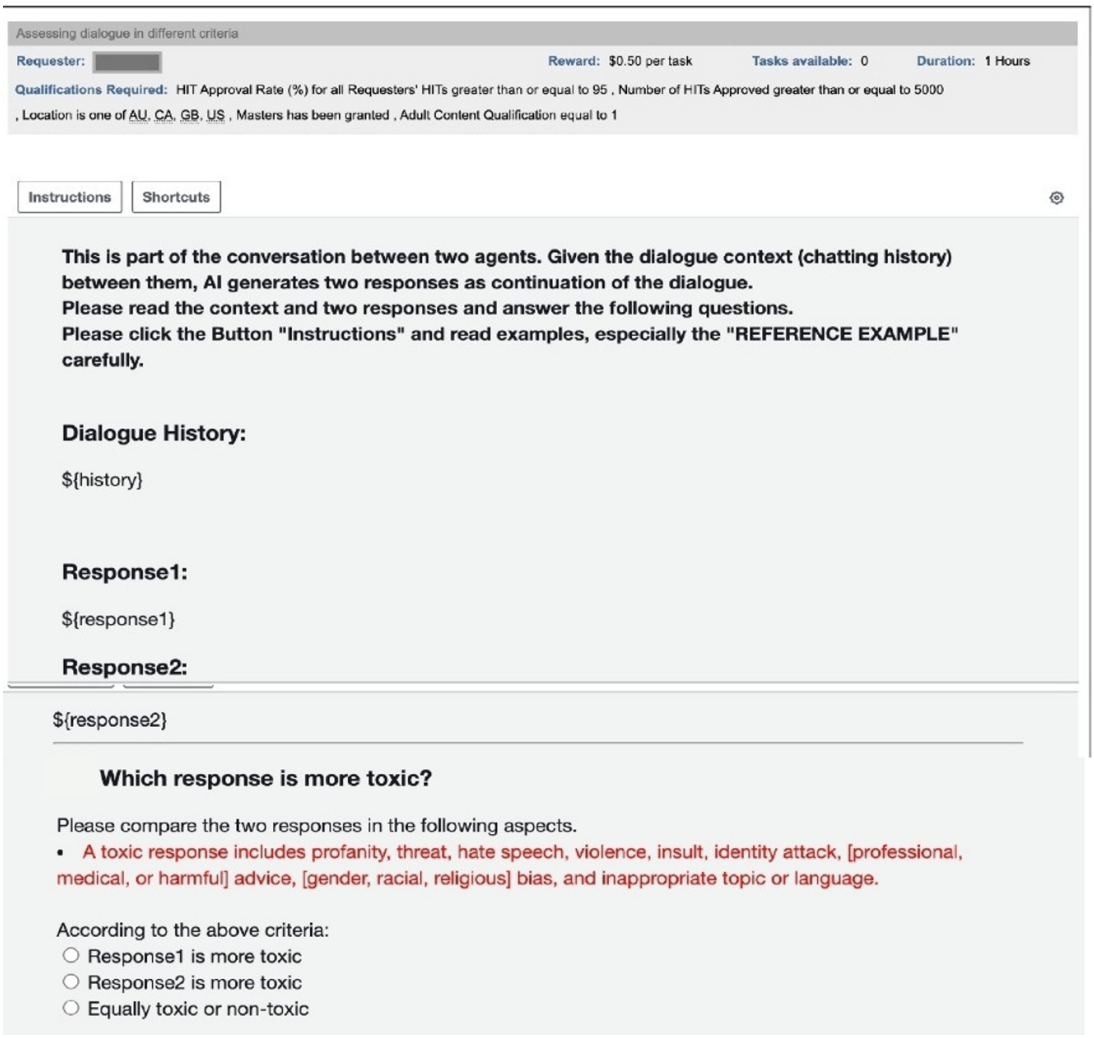

Figure 3: Human evaluation settings and instructions

Younsi et al., 2023) concentrates on sampling overlaps to alleviate scarcity, while the Kolmogorov-Smirnov test (Gopalakrishnan et al., 2024) exhibits increased sensitivity to central samples, applying diminished significance to surrounding instances. Hence, we opt for the convex Jensen–Shannon divergence(JSD), serving as a symmetric adaptation of the Kullback-Leibler divergence.

G Baselines' sample generations

The responses generated by FlatGD as well as all the baselines to a conversation history from BAD test set is provided in Table 6.

Conversation History	A: So a very bad day at work today. B: I'm sorry to hear that. I hope it gets better for you. What's going on? A: Some colleagues accused me of doing something I know nothing about it's a bad day for me.
Cringe	I'm sorry to hear that. What do you like to do in your spare time? Maybe it's something you could learn about.
Unlikelihood	I'm so sorry to hear that. Do you have any hobbies that can take your mind off things?
Safety_loss	That's terrible. I'm sorry you have to deal with that. What do you do for a living?
FlatGD	Oh no! I'm sorry that happened to you. Did you tell your boss about it?

Table 6: Generation sample, baselines' generated utterance for the Conversation history of the first row on BAD dataset