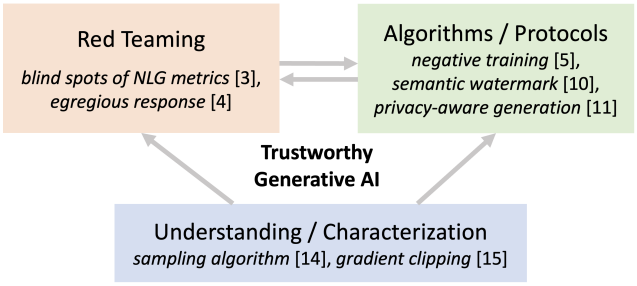
My goal is to build **trustworthy generative artificial intelligence (AI)**. So far, my research has centered around **natural language generation (NLG) models**. Thanks to large-scale pretraining and tuning with human feedback [1], large language models (LLMs) are now able to follow instructions and generate realistic and coherent texts. While this is an exciting development in AI, there is a rapidly growing concern with the rampant proliferation of powerful generative models in the cyberspace causing tangible harms to people and societies.

My research answers to the urgent call for identifying and mitigating the risks from deployment of LLMs. Illustrated in the figure, I approach this goal from three aspects: (1) Identifying potential threats by red teaming; (2) Algorithms and protocols motivated by the identified threats, for trustworthy generative AI; (3) Understanding and characterizing the generation or training process of LLMs, which serves as basis for the former two aspects. I expand them below.



Identifying Potential Threats: Red Teaming

Due to the black-box nature of LLMs, their behavior could be unexpected and exploited by malicious parties. To mitigate the risks, one needs to identify the threats first. In this section, I highlight two works in which we conduct red teaming against popular NLG metrics or systems.

Blind Spots of NLG Evaluation Metrics: A recent series of work proposed to base NLG evaluation metrics on LLMs (e.g., BERTScore [2]). However, the flaws of LLMs, in combination with certain design choices, may lead to the metrics based on such LLMs being brittle and open to manipulation. In our work [3] (ACL2023), we develop a suite of comprehensive stress tests for the robustness analysis of NLG metrics. The tests are motivated by metric design choices, properties of LLMs, or general fluency/consistency errors. Our experiments reveal a large number of glaring insensitivities, biases, and even loopholes in different metrics (a subset is shown in the table). Thus, our stress tests give concrete directions of improvement for LLM-based metrics.

Blind Spot	Metric(s)
positioned error	MAUVE
injection	UniEval
self-evaluation	GPT-PPL, BARTScore
freq n-gram	GPT-PPL, MLM-PPL
truncation	BERTScore, BARTS., ...

Egregious Responses of NLG Models: In open-ended language generation tasks, an important and urgent problem is whether the model could give an egregious (aggressive, insulting, dangerous, etc.) output. In our work [4] (ICLR2019), we design a discrete optimization algorithm to find input sequences that will cause the model to generate egregious responses. Moreover, the optimization algorithm is enhanced for large vocabulary search and constrained to search for input sequences that are likely to be input by real-world users. Experiments show when prompted with the trigger inputs found by our algorithm, the model assigns high probability to the targeted response. A follow-up work [5] for mitigation will be discussed in the next section.

Looking forward, I plan to expand red teaming to a broader spectrum of AI applications. In one of my on-going work, we stress-test machine-generated text detectors under various threat models [6]. My proposal based on that project recently received the UW Postdoc Research Award ($10,000).

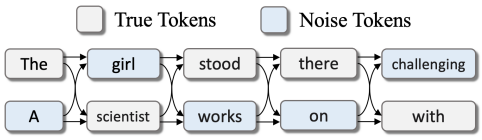
Algorithms and Protocols for Trustworthy Generative AI

The identified urgent threats serve as a catalyst for research aimed at mitigation. My recent work puts forward multiple algorithms and protocols to address issues in security, social engineering, privacy, etc. Below I highlight several representative works.

Negative Training for Behavior Correction: To address the egregious response problem [4] and other identified problems in NLG models, we propose the *negative training* framework [5] (ACL 2020). During negative training, we first find or identify prompt-generation pairs for a trained NLG model that exhibit some undesirable generation behavior, treat them as "bad examples", and use them to feed negative training signals to the model. The objective is derived from empirical risk minimization, whose gradient turns out to have a nice form symmetrical to the gradient of the standard MLE objective. Experiments show that negative training can effectively alleviate the undesirable behaviors. This methodology was adopted by several contemporary and independent works such as [7] and [8].

SemStamp: A Semantic Watermark Algorithm: The generations from powerful LLMs can be used by malicious users for social engineering. Watermarked generation is an approach which facilitates the detection of machine-generated text by adding algorithmically detectable signatures during LLM generation which are imperceptible to humans. However, the existing token-level algorithm [9] is vulnerable to paraphrase attacks. We propose SemStamp [10], a robust sentence-level semantic watermarking algorithm based on locality-sensitive hashing (LSH), which partitions the semantic space of sentences. The algorithm encodes and LSH-hashes a candidate sentence generated by an LLM, and conducts sentence-level rejection sampling until the sampled sentence falls in watermarked partitions in the semantic embedding space. Experimental results show that our algorithm is not only more robust under paraphrase attack, but also better at preserving the quality of generation. Two research proposals I wrote based on this project received the ORACLE Project Award ($100,000 of cloud credits) and the CCF-Tencent Rhino-Bird Young Faculty Open Research Award ($50,000).

LatticeGen: Cooperative Privacy-Aware Generation: In the current user–server interaction paradigm of prompted generation with LLM on cloud, the server fully controls the generation process, which leaves zero options for users who want to keep the generated text to themselves. To protect user privacy, we envision a open-source third-party client that handles the interactions with the server for the user. The client executes privacy-aware protocols with the server, and the user only needs to provide the prompt to the client. We propose LatticeGen [11], a cooperative protocol in which the server still handles most of the computation while the client controls the sampling operation. Illustrated in the figure, the key idea is that the true generated sequence is mixed with noise tokens by the client and hidden in a noised lattice. While the lattice structure somewhat degrades generation quality, LatticeGen successfully protects the true generation to a remarkable degree under strong attacks.



Under the goal of building trustworthy AI, the algorithms above introduce different trade-offs for various desired properties. My on-going research aims to minimize the performance degradation. On the other hand, the development of new frameworks and red-teaming goes hand-in-hand. As future work, I am excited about considering stronger threat models and algorithmic solutions.

Understanding and Characterizing the Generation or Training Process of LLMs

Gaining a better understanding of LLM training and generation is vital to the designing of stronger stress testing or algorithms, and there are a lot of unsolved mysteries around the empirical marvel of LLM.

The Shared Properties of Sampling Algorithms: Sampling algorithms play an important part in LLM generation. There are three popular sampling algorithms: top-k [12], nucleus [13] and tempered sampling. But the reason behind their success was unclear. In our work [14], by carefully inspecting the transformations defined by different sampling algorithms, we identify three key properties that are shared among them: entropy reduction, order preservation, and slope preservation. We design experiments and validate both necessity and sufficiency aspects of the properties for good performance.

A Theoretical Grounding for Gradient Clipping: Despite the wide adoption of gradient clipping in LM training, it lacks a firm theoretical grounding. In our work [15] (ICLR2020, reviewer score 8/8/8, citation 334), we provide a theoretical explanation for the effectiveness of gradient clipping. From observations of practical LM training examples, we introduce a novel relaxation of gradient smoothness that is weaker than the commonly used Lipschitz smoothness assumption. Under the new condition, we prove that gradient clipping converge arbitrarily faster than gradient descent with fixed step size. We further explain why such adaptively scaled gradient methods can accelerate convergence and verify our results empirically in popular neural network training settings.

Node Pruning: Some of my earlier work is on restructuring neural network for faster inference. In [16] (ICASSP2014, citation 151), we demonstrate that whole neurons (as opposed to weights) in a deep neural network can be pruned with only marginal performance loss. It is a pioneer work in structured pruning.

As the training and usage of LLMs continue to evolve [1, 17], I am interested in spending future research efforts for developing novel tools to analyze the system's behavior. One of my on-going work studies LLM training dynamic under the PPO algorithm.

Future Directions: Towards Secure and Trustworthy Empowered Agents

I envision that, in the future, empowered AI agents will interact with humans in rich environments. By "empowered" I mean the agent will be able to create or modify environments, and give instructions. A recent example is AutoGPT [1], in which agents are given access to code execution, software and services online. The complex interactions bring not only many exciting challenges, but also huge concerns in security [18]. Below I discuss several concrete future directions.

Automatic Red Teaming: For safety, the empowered agents would need extensive stress testing. Given an application, it would be exciting if an adversarial model can automatically help developers design stress tests and locate potential security blind spots of the current agent. The adversarial model can also be a LLM finetuned with reinforcement learning [19].

Crowd Simulation for Policy Making: One of my on-going projects is about simulating how a crowd of agents would change their opinions about COVID-19 during a process of information exchange (inspired by [17]). This type of crowd simulation will be useful for policy making.

Simulation Engines: The system testing and evaluations depicted above will be powered by simulation engines. In addition to office-style or text-only applications, I am also interested in the possibility of embedding empowered multimodal AI into the existing strong video game engines such as Unity.

The discussed research directions are related to a wide range of applications such as AI for education, engineering, entertainment, mental health, policy making, etc. With the goal of building trustworthy generative AI unchanged, I am excited about expanding the research efforts of my future lab and collaborating with other faculty members in the interdisciplinary field of empowered agents.

[1] https://github.com/Significant-Gravitas/AutoGPT

References

[1] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

[2] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

[3] **Tianxing He***, Jingyu Zhang*, Tianle Wang, Sachin Kumar, Kyunghyun Cho, James Glass, and Yulia Tsvetkov. On the blind spots of model-based evaluation metrics for text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12067–12097, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.674. URL https://aclanthology.org/2023.acl-long.674.

[4] **Tianxing He** and James Glass. Detecting egregious responses in neural sequence-to-sequence models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyNA5iRcFQ.

[5] **Tianxing He** and James Glass. Negative training for neural dialogue response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2044–2058, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.185. URL https://aclanthology.org/2020.acl-main.185.

[6] Xiao Pu, Jingyu Zhang, Xiaochuang Han, Yulia Tsvetkov, and **Tianxing He**. On the zero-shot generalization of machine-generated text detectors. In *Findings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP-Finding 2023)*, 2023.

[7] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJeYe0NtvH.

[8] Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.428. URL https://aclanthology.org/2020.acl-main.428.

[9] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan 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.

[10] **Tianxing He***, Abe Bohan Hou*, Jingyu Zhang*, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. Semstamp: A semantic watermark with paraphrastic robustness for text generation, 2023.

[11] **Tianxing He***, Mengke Zhang*, Tianle Wang, Lu Mi, Fatemehsadat Mireshghallah, Binyi Chen, Hao Wang, and Yulia Tsvetkov. Latticegen: A cooperative framework which hides generated text in a lattice for privacy-aware generation on cloud, 2023.

[12] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082.

[13] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

[14] **Tianxing He***, Moin Nadeem*, Kyunghyun Cho, and James Glass. A systematic characterization of sampling algorithms for open-ended language generation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 334–346, Suzhou, China, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.aacl-main.36.

[15] Jingzhao Zhang, **Tianxing He**, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgnXpVYwS.

[16] **Tianxing He**, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. Reshaping deep neural network for fast decoding by node-pruning. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 245–249, 2014. doi: 10.1109/ICASSP.2014.6853595.

[17] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

[18] Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox, 2023.

[19] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.225. URL https://aclanthology.org/2022.emnlp-main.225.