Language Models (LMs) have become a cornerstone of modern AI systems. Domain-Specific Language Models (DSLMs) refer to the specialized models that are trained for specific sensitive domains, such as medical diagnosis, financial analysis, and legal advice. DSLMs are very important because they can provide more accurate and efficient services than the general LMs. It is estimated that DSLMs will take over 60% of the enterprise models by 2028. Given the sensitive nature of DSLMs, the model security of DSLMs is important due to the scarcity of the high-quality data required to train the models and the privacy-sensitive nature of the data. For example, in medical diagnosis, the model owner must collect and label real data from the patients such as X-rays, CT scans, and medical records. These data are very sensitive and cannot be leaked to the outside world. Leaking the model parameters can leak these sensitive data and cause severe consequences. Confidential platform is important for future DSLM as it is estimated to take over 70% of the DSLM workload by 2029. My research aims to **build a confidential platform to protect the model security of DSLMs**. This platform can store and process sensitive DSLM weight parameters in a secure and confidential manner for DSLM applications. It is difficult for existing solutions to protect DSLM privacy without introducing large cost. Cryptography techniques can increase the computation overhead by thousands of times. Differential privacy decreases the model accuracy by a large margin. Federated learning cannot provide strong privacy guarantees on the model weights.

I designed a TEE-based confidential heterogeneous computation platforms, `TAOISM` [8], which combines trust execution environments (TEEs) and widely deployed non-TEE GPUs. TEE is a seperated area of memory and CPU that's protected from the rest of the machine and can provide strong security guarantee on the code execution and data confidentiality. My insight is to **partition the DSLM into a privacy-critical part and a computation-intensive part, and deploy each part in the appropriate environment**. The privacy-critical part is protected by TEE and the computation-intensive part is offloaded to non-TEE GPUs. TEE has a stronger security guarantee but limited computation capability, while non-TEE GPUs have high computation capability but do not have protection schemes. By carefully designing the partition strategy and properly protect the offloaded part, I can achieve a better trade-off between security and utility and provide a more secure and efficient AI service. **`TAOISM` has received the Outstanding Doctoral Dissertation Award of Peking University.** My research is at the frontier of LM security and trusted hardware, and can ensure the privacy and efficiency of LM data. Centered on `TAOISM`, my research consists of three parts:

- First, I design a secure and efficient model deployment framework for DSLM. This framework can analyze the neuron-wise importance [9], partition the DSLM based on different levels of security requirements [8, 5], and obfuscate the model parameters on non-TEE GPUs [6, 11]. **The framework has been deployed in the AI confidential computing service of Bytedance.**

- Second, I design a interface audit framework to protect the input/output data of the TEE-shielded DSLM, which is beyond the existing TEE's protection scope. This framework can measure training data privacy leakage [10], identify the security vulnerability of TEE interface [7], and audit the model structure leakage [4].

- Third, I design a set of LM-enhanced and system-level defenses based on `TAOISM` to improve the security of commercial AI products, including an on-device user authentication system [1] and a provenance analysis defense against adversarial attacks [3]. **The authentication solution has been used in Alipay and influence over ten millions of users.**

Heterogeneous DSLM deployment framework based on TEE

TEEs and colocated non-TEE GPUs can provide different and complementary capabilities for secure and efficient DSLM deployment. To coordinate the advantages of both, I design a heterogeneous DSLM deployment framework to utilize the double benefits of TEE's high security guarantee and non-TEE GPUs' high computation capability. The framework first partitions the DSLM and then deploy the partitioned parts in the appropriate environment.

My research, NNSlicer [9], can analyze the neuron-wise sensitivity and importance of the DNN model and select the most sensitive neurons to protect. Motivated by dynamic slicing in program analysis, NNSlicer computes a subset of neurons and synapses that may significantly affect the values of certain interested neurons and utiliize TEEs to efficiently protect model functionality. However, recently the wide existence of pre-trained LMs enhance the capability of model stealing attackers because they have more knowledge about the victim model. The attacker can utilize the public pre-trained model as a prior knowledge and utilize the model part on the non-TEE GPUs to partially fine-tune the model to recover the full functionality with low cost. My research systematicallyt study the defense effectiveness of prior TEE-based protection methods under the stronger adversary and reveal the limitations of existing methods. The defense effectiveness of these methods is similar to no defense. I summarize the foundamental limitation is the training-before-partitioning strategy, which means that the model is first trained by private data as a whole and then partitioned into different parts. Because the training phase updates all parameters, it is difficult for the defender to isolate the private parameters from other parameters. Motivated by this limitation, I design TEESlice [8] that utilizes the partition-before-training strategy to isolate the model privacy from the public pre-trained model. My strategy is to partition the model into a privacy-related part and a privacy-irrelevant part. During training the private training data only updates the privacy-related part and does not update the other part. The privacy-irrelevant part is deployed to the non-TEE GPU and, even under the stronger adversary, the attacker cannot recover the functionality of the protected model. Another limitation of existing model partition solutions is the communication cost between TEE and non-TEE GPUs, which can cost up to 40% of the total inference time. To address this, I design AegisGuard [5] to selectively shield the sensitive light-weight adapters in TEE. AegisGuard utilize reinforcement learning to measure the layer sensitivity and actively reduce the shielded adapters.

My research also design lightweight obfuscation algorithms for the model parameters on non-TEE GPUs. Because the non-TEE GPUs are fully exposed to the attacker, he can utilize the offloaded model parameters to recover partial functionality of the victim model. Thus, the model parameters on the GPU should be properly protected meanwhile not harming the model inference efficiency. My research demonstrated that, if the obfuscation algorithm is not properly designed, the attacker can easily utilize public pre-trained models to recover the model parameters [8]. To address this, I propose two obfuscation schemes, GroupCover [11] and GameofArrows [6], each of which resists such attacks under distinct threat models and with minimal runtime overhead. Specifically, GroupCover protects against vector alignment attacks by grouping and linearly combining weight vectors, thereby hiding one-to-one correspondences between private and public weight vectors. By forming each obfuscated vector as a randomized linear combination of multiple original vectors, GroupCover breaks direct matching between private and public models while preserving the ability to reconstruct correct inference through an efficient decoding transformation inside the trusted environment. GameofArrows applies a random matrix–vector multiplication in the TEE before releasing weights to the GPU. This operation subtly but effectively disturbs the directional signatures of each weight vector, making angular matching unreliable. Critically, GameofArrows remains computationally lightweight: matrix–vector multiplication is far cheaper (over two orders of magnitude) than full matrix–matrix computation of the linear layers. I further demonstrate that the protection scheme of GameofArrows can be reduced to a secure cryptographic primitive, learning-with-errors (LWE), which theoretically demonstrate the difficulty of recovering the model parameters from the obfuscated model. In summary, GroupCover and GameofArrows complement each other: GroupCover hides vector correspondences, and GameofArrows obfuscates vector direction. Together, they form a robust defense suite for lightweight parameter protection on GPUs, resisting adversarial use of public pre-trained models while preserving inference performance.

Audit framework for TEE-shielded DSLM

TEE cannot protect the output interfaces of the software. For DSLM, the model output can leak the model information and be used to infer model privacy. To address this problem, TAOISM consists of a set of audit tools to measure the information leakage and security threats from the output interfaces. My research, SymGX, detects cross-boundary pointer vulnerabilities in SGX applications where data is exchanged via ECalls and OCalls. It builds a new SGX-specific analysis model, Global State Transition Graph with Context-Aware Pointers (GSTG-CAP), to accurately simulate multi-entry execution, stateful global variables, and pointer semantics. This model enables SymGX to systematically audit enclave interfaces and identify insecure pointer usage that may lead to sensitive memory exposure. It is a common practice to use knowledge distillation (KD) and deploy a smaller student model in the TEE to reduce the computation workload. My research, PPKD, systematically study membership information leakage and memorization risks in knowledge distillation, which can be exploited by the attacker to gain the privacy of the training data. My research reveals that existing KD methods are not enough to protect the training data privacy and provides a guideline to design a safer TEE-oriented KD algorithm that can minimize the privacy exposure through model outputs. My research ModelDiff audits the risk of model family leakage from the model inference output. It computes model output similarity to detect knowledge reuse and unintended inheritance of insecure logic. It introduces decision distance vectors (DDVs) to compare models' behavioral patterns over shared inputs in a black-box setting. By measuring semantic output similarity, ModelDiff identifies unauthorized model reuse or vulnerability propagation through external I/O interfaces.

AI-enhanced defenses based on TAOISM for commercial products

Based on TAOISM, I design several AI-enhanced defenses for commercial systems to reduce the privacy leakage and enhance the security of the products. FAMOS [1] embeds an adversarial filtering mechanism within the TEE so that malicious input perturbations are preemptively sanitized before reaching the model, reducing attack success rates substantially. To fit the limited computation resource of TEE, I design a novel residual model architecture that can effectively project the multi-modal user features to a low-dimensional representation. This design maintains low latency overhead, making it viable for responsive production systems on smartphones. QPA [3] (Query Provenance Analysis) tracks the historical relationships among client queries by constructing a provenance graph inside the TEE, distinguishing benign query sequences from malicious ones. Through anomaly detection on these provenance structures, QPA rejects probing or attack queries while preserving high throughput.

Ongoing and short-term future work

Security agent benchmarking. LLM agents have shown promising capabilities in software engineering tasks and can achieve over 70% accuracy on the SWE-bench. This draws a great potential to build security agents to work on the security tasks. However, existing security benchmarks are not comprehensive for real-world security tasks. My recent research, Sec-Bench [2], takes the first step to evaluate the capability of LLM agents on real-world and authentic software security tasks. My research demonstrate the large space of opportunities for vulnerability detection, patch generation, proof-of-concept input generation, which can be completely automated by LLM agents. Sec-Bench also reveals soveral limitations of existing LLM agents and provides a guideline for future research, such as context management, reasoning ability on complex vulnerabilities, and the planing capability to generate complex inputs.

TEE-based data sharing platform for domain-specific LLMs. The large computation and storage cost of domain-specific LLMs requires the collaboration of multiple parties. However, for areas that require high-quality data and privacy-sensitive data, the existing data sharing platforms cannot provide the security guarantee. When the users want to share the data, they require the security guarantee on the data sharing process. My research, TAOISM, have designed a secure computation platform for domain-specific LLMs, which can provide the security guarantee for the models. The model partition design and data obfuscation algorithms in TAOISM can be extended to protect a broader range of data sharing scenarios in finance, healthcare, and other sensitive domains.

Protecting agent assistants by TEE-based tool chain. The agent assistants are widely used in the software development process. However, the security of the agent assistants is not well-studied. Existing research have demonstrated that LLM agents are vulnerable to various prompt injection attacks that are initiated from untrusted sources. My research, TAOISM, can be extended to protect the security of agent assistants by only accepting the data from trusted sources. This can be achieved by deploying the tools in the TEE and ask the TEE to sign the tool output. Agents can verify the data signature and distinguish the trusted tools from the untrusted ones to avoid the prompt injection attacks.

Long-term future work

Automatic cybersecurity agents. I also envision to build a set of fully automated cybersecurity agents that can continuously detect, analyze, and repair software vulnerabilities without human intervention. The cybersecurity tasks are complex for existing LLMs because they cannot handle the intrinsic logics of the software vulnerabilities. With the rapid development of multi-agent collaborative agents, we now have the foundation to create agent frameworks capable of understanding system semantics, generating exploits, and synthesizing defenses in a closed adaptive loop. My long-term goal is to develop a self-evolving security ecosystem where AI agents serve as both builders and protectors of the digital infrastructure, integrating symbolic reasoning, reinforcement learning, and trusted execution to ensure verifiable and autonomous defense. Ultimately, Automatic Cybersecurity Agents represent a paradigm shift—from reactive protection to proactive, self-healing security that keeps pace with the speed of modern computing.

Secure agentic systems. As AI agents become interconnected through shared tools, APIs, and memory contexts, new classes of attacks—such as prompt injection, model context poisoning (MCP), and cross-agent manipulation—emerge, threatening the reliability of autonomous workflows. I aim to develop principled defenses that combine context integrity verification, tool interaction verification, adaptive trust modeling, and TEE-based isolation to secure agent collaboration at scale. Ultimately, this direction seeks to establish the foundation for trustworthy multi-agent intelligence, where large collections of autonomous agents can cooperate safely, transparently, and resiliently in open ecosystems.

Security system for embodied intelligence. As intelligence begins to inhabit the physical world, the security of embodied intelligence will define the next frontier of trustworthy AI. Embodied agents interact with a more complex and dynamic physical environment and are exposed to a wider range of physical threats. My goal is to build a security framework for embodied intelligence, including drones, autonomous vehicles, and robots, to ensure the safety and reliability of their behaviors. Based on TAOISM, I can isolate the core functionality of the embodied intelligence and protect this part from the external threats.

References

[1] Y. Cai, Z. Zhang, M. Yao, J. Liu, X. Zhao, X. Fu, R. Li, Z. Liu, X. Chen, Y. Guo, et al. I Can Tell Your Secrets: Inferring Privacy Attributes from Mini-app Interaction History in Super-apps. In *34th USENIX Security Symposium*, 2025.

[2] H. Lee, Z. Zhang, H. Lu, and L. Zhang. SEC-bench: Automated Benchmarking of LLM Agents on Real-World Software Security Tasks. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025.

[3] S. Li, Z. Zhang, H. Jia, Y. Guo, X. Chen, and D. Li. Query Provenance Analysis: Efficient and Robust Defense against Query-based Black-box Attacks. In *2025 IEEE Symposium on Security and Privacy*, 2025.

[4] Y. Li, Z. Zhang, B. Liu, Z. Yang, and Y. Liu. ModelDiff: Testing-based DNN Similarity Comparison for Model Reuse Detection. In *30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021.

[5] C. Wang, Z. Zhang, Y. Wang, T. Wang, Y. Hao, J. Gao, T. Wei, Y. Cao, Z. Chen, and W. Lim. AegisGuard: RL-Guided Adapter Tuning for TEE-Based Efficient and Secure On-Device Inference. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025.

[6] P. Wang, B. Dong, Y. Cai, Z. Zhang, J. Liu, H. Xue, Y. Wu, Y. Zhang, and Z. Zhang. Game of Arrows: On the (In-Security) of Weight Obfuscation for On-Device TEE-Shielded LLM Partition Algorithms. In *Proceedings of the 34th USENIX Conference on Security Symposium*, 2025.

[7] Y. Wang, Z. Zhang, N. He, Z. Zhong, S. Guo, Q. Bao, D. Li, Y. Guo, and X. Chen. SymGX: Detecting Cross-boundary Pointer Vulnerabilities of SGX Applications via Static Symbolic Execution. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023.

[8] Z. Zhang, C. Gong, Y. Cai, Y. Yuan, B. Liu, D. Li, Y. Guo, and X. Chen. No Privacy Left Outside: On the (In-) Security of TEE-Shielded DNN Partition for On-Device ML. In *2024 IEEE Symposium on Security and Privacy*, 2024.

[9] Z. Zhang, Y. Li, Y. Guo, X. Chen, and Y. Liu. Dynamic Slicing for Deep Neural Networks. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020.

[10] Z. Zhang, A. S. Shamsabadi, H. Lu, Y. Cai, and H. Haddadi. Membership and Memorization in LLM Knowledge Distillation. *CoRR*, abs/2508.07054, 2025.

[11] Z. Zhang, N. Wang, Z. Zhang, Y. Zhang, T. Zhang, J. Liu, and Y. Wu. GroupCover: A Secure, Efficient and Scalable Inference Framework for On-Device Model Protection Based on TEEs. In *Forty-first International Conference on Machine Learning*, 2024.