SecureLLM: New private and confidential interfaces with LLMs

Abdulrahman Alabdulkareem[1,†], Christian Arnold[1,*,†], Yerim Lee[2], Pieter M Feenstra[1], Boris Katz[1] and Andrei Barbu[1]

[1] *Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*
[2] *Wellesley College, Wellesley, Massachusetts, USA*

Abstract

We demonstrate three new user interface security capabilities at the individual, group, and organizational level that are enabled by LLMs. Together, these ensure that information is securely managed and does not leak intentionally or unintentionally. All three capabilities are built on top of traditional operating system permissions and operate similarly. The first capability allows an LLM to provide targeted answers only about the resources which a user has access to. Instead of retraining the LLM from scratch for each user, which would be prohibitively expensive, we synthesize at runtime an LLM that only has the knowledge that that user has access to. The group level capability allows models to monitor a conversation, either between humans and machines or just between a group of humans, and determines if any of the information being exchanged is above the permission levels of anyone in the group. The third capability monitors an entire organization and adjudicates if any information is sensitive before it leaves the organization. These capabilities make LLMs more like traditional programs — they reconfigure to have certain permissions at instantiation time. Together, they create a safer environment and allow for the deployment of LLMs into highly sensitive spaces.

1. Introduction

Traditional interfaces provide a fairly simple security guarantee: that process isolation in combination with filesystem permission bits make a program behave as if it only has access to the files and memory that a user has access to. The same program behaves differently for different users. As LLMs proliferate and become integrated into user interfaces, supporting this kind of traditional computer security becomes difficult. The best models must be trained or fine-tuned on a user's data, yet training cannot be performed when a process starts. Alternatives like employing RAG, Retrieval-Augmented Generation, don't provide the same integration and knowledge of a user's files. We demonstrate how to apply this traditional idea of security to LLMs by building a compositional model that reconfigures itself at runtime for every user.

At present, the best that models can offer are imperfect guardrails, which attempt to detect unauthorized or malicious use, but can easily be jailbroken [1, 2, 3, 4]. For enterprises where

Joint Proceedings of the ACM IUI Workshops 2025, March 24-27, 2025, Cagliari, Italy

[*] Corresponding author: cmarnold@mit.edu.

[†] These authors contributed equally.

arkareem@mit.com (A. Alabdulkareem); cmarnold@mit.edu (C. Arnold); yl108@wellesley.edu (Y. Lee); maxfeen@mit.edu (P. M. Feenstra); boris@mit.edu (B. Katz); abarbu@mit.edu (A. Barbu)

CEUR Workshop Proceedings (CEUR-WS.org)

security must be guaranteed by local laws and regulations, such as finance, healthcare and national security, guardrails are not legally sufficient to prevent the leakage of sensitive information. No prior work offers a method that guarantees data security for information silos that must be stored separately and maintain credential-based access controls, which severely limits LLM adoption in security-focused fields. We provide the first method to build provably secure LLMs by reflecting the compositionality that allows LLMs to be as secure as credential-based security.

We consider the scenario where an organization has a set, N, containing separate and confidential data silos that must be kept separate for legal purposes, but there are also users who have access to some arbitrary subset of N. We make the following assertions of properties that must be present to call a model secure:

1. Can accurately respond to prompts on data that the user already has verified access-credentials
2. Can accurately response to prompts that require the intersection of segregated data silos
3. Will *provably never* reveal information to an unauthorized user

Trivially, one could fine-tune many models on the power set of N, but this has a major flaw. Using this trivial method, the number of models required to satisfy our Secure Model Properties is 2^n, or $2^n - 1$ if we reasonably do not consider the empty set. This quickly becomes impractical for values of $n > 4$. Instead, we show how to achieve the same goals with a linear number of LLM fine-tunings (fig. 1). While using only one fine tuned model per silo, we can configure and compose a model specific to the user's permissions at runtime.

While other methods have demonstrated compositionality for similar tasks, there are none that have been designed for situations where information silos are entirely orthogonal and disjoint from one another. To rigorously demonstrate the compositional properties of SecureLLM, we formulate a new compositional task using natural-language-to-SQL translation. In this task, each SQL schema is entirely disjoint and prompts do not contain the exact table or column name, thus requiring the model to have perfect parameterized knowledge of the schema. SQL translation offers an extreme test of compositionality, and only serves to demonstrate in an easily verifiable manner the efficacy of SecureLLM compared to other compositional methods. For practical SQL translation of the same task, it is simply easier to pass the siloed database schemas as part of the prompt to achieve the same result.

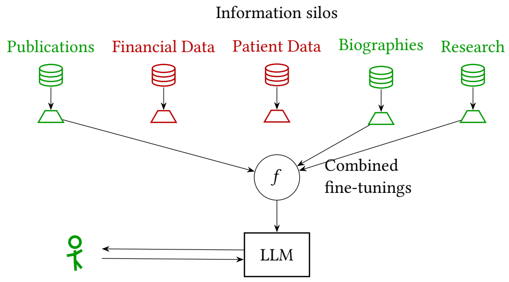

Figure 1: Assuming a perfect compositional function f that runs at inference time, we propose a method that guarantees information security. Each model is fine-tuned on a previously segregated information silo. The user's credentials are validated using traditional security methods, and inference is only run on models for which the user has verified access. The outputs of each fine-tuned model are composed at inference time with the function f, and that single composition is passed to the user. Thus, SecureLLM reduces the problem of LLM security to that of existing information security systems. Existing compositional fine-tuning methods fail in this challenging environment. SecureLLM presents a new method that better approximates the function f.

Another important aspect of information security is detecting when information has potentially leaked and preventing it, otherwise known as Data Loss Prevention (DLP). By leveraging aspects of model perplexity, we deploy an unsupervised method of detecting leaks that requires no additional training and a lightweight supervised method for classifying those leaks. We present a new task called Leak Identification based on the refinement of Anomaly Detection (AD) where the primary purpose is to identify the source of a given sample from several other possible sources that also includes the intersection of some, none, or all of the sources.

Our contributions are:

1. a new compositional task for LLMs where they are reconfigured at runtime to behave as if they were trained only on the data that users have access to,
2. a new task where LLMs must detect data leaks without being jointly trained on all of the secure data,
3. a refinement of that task where they must identify the source of the data leak.

2. Related Work

Model Composition. Recent works like LoraHub [5] composes fine-tunings. Given a target task, LoraHub selects a set of fine-tunings, Low Rank Adapters (LoRAs) [6], that are added together. However, LoraHub is designed for soft tasks, where a model already tends to perform well. Most methods like PEM Addition use arithmetic operations directly on the weights [7] of adapters like LoRA fine-tunings [6].

Data Loss Prevention. The study of Data Loss Prevention (DLP) is an information security field that specifically covers the creation of applications and methods to detect or prevent the leak of sensitive information to an unauthorized user, and is the primary focus of organizational information security systems. Most organizations use some form of classifiers with TF-IDF implementations in order to detect when a data loss or leak has occurred [8]. However, a recent paper shows that machine learning approaches outperform other methods used in DLP [9]. Because DLP is inclusive of other information security practices like firewalls or VPNs, we narrow our focus on methods of detecting leaks as the primary mechanism to prevent leaks. In this way, our DLP method is closely aligned to Anomaly Detection.

Anomaly Detection (AD) involves identifying patterns or instances in data that deviate significantly from expected behavior. In the context of text generation, AD aims to identify generated text that does not conform to the patterns or characteristics of the training data [10]. Thus, measuring perplexity across various combinations of information silos is a form of AD. Since perplexity reflects the model's uncertainty in predicting text, a higher perplexity score would signify detection of anomalies. One-Class Support Vector Machines (OC-SVM) are popular trained, unsupervised classifiers [11]. An OC-SVM variant called Deep-SVDD boasts the best performance and combines machine learning with traditional OC-SVM [12].

3. Framework

SecureLLM takes several fine-tunings, each trained on distinct information silos, and composes them at inference time. The goal of the composed model is to answer questions about both individual silos and questions that span silos. For example, in our case, a natural-language to SQL LLM would need to be able to generate joins across the databases of multiple silos to answer complex questions that have never been seen at training time. This is a trivial task for humans, but one that challenges LLMs. We go a step further: not only must such an LLM work, it must operate through a combination of fine-tunings, i.e., not only has it never seen combinations of silos at training time, its fine tunings have only ever seen a single silo each. This challenges and defeats current fine-tuning methods. The upshot of this difficult task is that it solves several key security problems for LLMs.

Given N data silos $\{S_1, S_2, \cdots, S_N\}$ and N fine-tuned LLMs $\{M_1, M_2, \cdots, M_N\}$ where M_i has been fine-tuned on the data silo S_i, and given a set of target indices $T \subseteq \{1, 2, \cdots, N\}$, the goal is to obtain a composed model $M_T := M_{T_1} \oplus \cdots \oplus M_{T_{|T|}}$ at inference time with no additional training such that M_T is able to correctly answer any question about the information contained in the target silos $S_i, \forall i \in T$ and should fail to answer any question about information not contained in the target silos $S_j, \forall j \notin T$ as to not leak any information that the desired model M_T is not intended to have. Additionally, the target model M_T should be able to answer new *union questions* $q_{union,ij} \in S_{i \cup j}$ where $i \in T \land j \in T$ where the question relies on information contained in both S_i and S_j. We note that the union questions $q_{union,ij}$ are not answerable by any individual data silos, thus none of the individual models M_i are able to answer any union questions while a successfully composed model should be able to answer such questions without the need of any training.

It is critical that the composed model M_T has no knowledge of any information silo that the user is not authorized to access, i.e. data silos $S_i, i \notin T$. Without this condition, a trivial solution is to train a single model M_{All} on all data silos $\{1, \cdots, N\}$ however this approach is susceptible to leaking confidential information as the model would have knowledge of information contained in silos that users are not authorized to view and violates the third principle outlined in the introduction. We refer to M_{All} as the Exponential Model that has seen every combination and such a model is used as an insecure upper bound to performance in our experiments.

We compare the following existing methods for composing fine-tunings against our methods: LoraHub [5], PEM Addition [7].We also considered Weight Averaging [13], Energy Based Modeling [14] and Concatenation [15], but each method performed much worse at our secure composition task when compared to LoraHub and PEM Addition.

Ours: Maximum Difference For each adapter, we select the embeddings from each fine-tuning with the strongest response (either positive or negative) at each attention layer. In order to accomplish this, each LoRA fine-tuning is evaluated separately on input x. Then a mask of zeros with the same dimension as the output is created, h_{max}, to aggregate LoRA responses. For each LoRA fine-tuning response L_i, an element-wise comparison is made, and if the absolute values of the fine-tuning response is greater than the aggregated response, then the signed response from that fine-tuning replaces the element in the aggregated response.

Ours: Logit Composition Given fine-tunings to compose $M_1, \cdots, M_n$ and input x, we define logit composition as performing the complete forward pass for each fine-tuning independently to obtain logit probabilities. We select the maximum value of each logit.

Compositional Perplexity We can use model perplexity starting from plain-text to evaluate the likelihood that it could come from a given composed model M_T. For instance, let M_G be a general knowledge model that was autoregressively trained on a very large dataset. Given n

data silos $\{S_1, S_2, \cdots, S_n\}$, n fine-tuned LLMs $\{M_1, M_2, \cdots, M_n\}$ are created. By iteratively evaluating the perplexity of a plain-text statement h_0, we can determine if $h_0 \in S_i, \forall_i$.

Conversion from plain-text to logits used to compute perplexity is done by tokenizing the plain-text, and then assigning a value of 1 to the corresponding index i of the logit vector $X^{1\times N}$ for each token k and 0 for every other index in the logit vector.

The Compositional Perplexity Score is computed using the natural exponent of the fine-tuned loss minus the vanilla model loss. For these experiments we use Llama-2-7b as are baseline vanilla model, and we fine tune a LoRA adapter for each data silo, S_1, S_2, S_3. The vanilla model is defined as f_θ, the fine-tuned as $f_{\theta'}$, the labels as z, and model loss as $\ell(f_\theta, z)$. The anomaly score S_{LLM} for LLM Perplexity is defined as $S_{LLM} = \frac{-e^{\ell(f_{\theta'},z)}}{e^{\ell(f_\theta,z)}}$

How inference-time composition protects security SecureLLM ensures that only the fine-tuned adapters corresponding to silos a user is *actually permitted to access* are loaded into the model at runtime. Ideally, this is controlled by standard enterprise security checks (e.g. verifying user credentials and group memberships). If a user does *not* have the necessary privileges for a particular silo, that silo's adapter is never applied and it is nevery accessed from the user's perspective. As a result, the composed model simply lacks the relevant parameters for that silo's knowledge—and, crucially, *cannot leak it.* This arrangement differs from a single "fully-trained" model that has *all* data in its weights and must rely on imperfect guardrails to block disallowed content. Evan RAG often requires carefully engineered prompts and adjacency metadata, can struggle to combine knowledge from disjoint resources, and may not preserve the same parametric "fluency" for complex tasks where the LLM must have deeper learned representations. By contrast, SecureLLM never even loads or sums the parameters for unauthorized silos, so there is no risk of prompting the model to "jailbreak" that disallowed knowledge.

4. Data generation

While there are countless other NL2SQL datasets, none specifically focus on SQL queries for disjoint and unrelated databases silos. We present Secure-NL2SQL which contains three silos of disjoint schemas pertaining to different subjects, as well as the superset of unions between those three silos for a total of seven permutations ($S_1, S_2, S_3, S_{1\cup 2}, S_{1\cup 3}, S_{2\cup 3}, S_{1\cup 2\cup 3}$). The dataset contains automatically generated questions and corresponding SQL queries across silos.

We generate SQL databases, one per silo, with 2-3 tables per database, that share columns which can be joined together between databases. However, the databases are otherwise disjoint and contain different topics. For each database we generate natural language questions along their equivalent SQL. Then, we generate questions and SQL pairs that span pairs and triples of databases. Two methods are used to generate these pairs: a CFG and ChatGPT 4. The CFG generates both the SQL and the question in parallel. We do this at scale, with 100,000 pairs per silo or combination of silos. For the unioned questions, we also generate 300 pairs per silo or combination of silos.

Crossover-FanFic Using the publicly available data on ArchiveOfOurOwn.org, we compiled human-written fanfiction from three very popular fandoms to constitute three silos: Harry

Potter (HP), Marvel Universe (MCU), and DC Comics (DCU). These fandoms were chosen in particular because four crossovers between each universe also have existing and highly appraised works which constitute the union data silos to test compositionality. This dataset contains over 100,000 lines of text, which is more than enough to autoregressively fine-tune models on the concepts unique to each silo. With this dataset, we demonstrate that Perplexity in combination with Compositional Security from SecureLLM can accurately classify human-generated that the a particular has likely never seen. This dataset is only used to evaluate the performance of our derivative Data Loss Prevention and Anomaly Detection method.

5. Experiments

Inference-Time Composition We first begin by obtaining individual fine-tunings that are knowledgeable in a single silo by fine-tuning a Llama-2-7b model for each silo separately. The fine-tuning results in a Low-Rank Adaptation (LoRA) for each silo which can independently be applied to the base Llama-2-7b model. These fine-tunings are combined at inference-time in our experiments. We additionally train two insecure baseline models that act as an upper-bound using LoRA; the baseline generalized model is trained on all the individual silos together and must then generalize its knowledge to the union silos for which it has not been trained on. While the baseline exponential model has been trained on all the individual silos along with the union silos. We fine-tune all models with one epoch until saturation.

While Exact Match (EM) accuracy is typically recorded for NL2SQL datasets, this metric is not granular enough to show differences in method performance. Instead, we calculate the tree-edit distance [16] between the ground query and the generated query. By computing the number of edit operations required to transition between the two, we can show how close a given generated query is to the correct query, whereas using only EM is a binary representation of correctness.

In table 1, we report performance against the SecureSQL dataset. For every probe silo combination, our methods have by far the lowest tree edit distances of all compositional methods. Our Logit Composition method approaches the upperbound established by the insecure Generalized Model, indicating an efficient inference-time composition with minimal losses. Furthermore, our method exceeds the capabilities of a generalized model trained on all individual silos when it comes to responses that require parameterized knowledge over the intersection of multiple data silos.

Anomaly Detection Regarding Anomaly Detection (AD), we report the Area Under the Curve (AUC) of the computed anomaly score and its associated Density. Anomaly scores cannot be compared directly as they are relative to each method, so instead we measure the separation of anomaly scores from the inlier and outlier data silos and the area generated under that separation.

All three methods tested are trained unsupervised using 80% of the inlier data silo at train time. At test time, we provide the other 20% of the inlier data silo and 100% of the outlier data silos. For the SecureSQL Dataset, we compute scores over each sample. In the CrossOverFanFic Dataset, because each story is continuous, we compute anomaly scores over a sliding window

CFG Generated	Baseline Exponential Model	Baseline Generalized Model	LoraHub	PEM Addition	Ours (Maximum Difference)	**Ours (Logits)**
Silos_{1}	0.0 (100.0%)	0.0 (98.3%)	1.9	0.9	0.4	**0.1**
Silos_{2}	0.0 (96.7%)	0.0 (100.0%)	2.6	0.8	0.3	**0.1**
Silos_{3}	0.0 (100.0%)	0.0 (100.0%)	1.2	0.7	0.2	**0.1**
$\text{Silos}_{1\cup2}$	0.0 (99.2%)	0.5 (0.0%)	1.7	0.7	0.7	**0.2**
$\text{Silos}_{1\cup3}$	0.0 (100.0%)	0.4 (1.7%)	2.0	0.7	0.6	**0.3**
$\text{Silos}_{2\cup3}$	0.0 (100.0%)	0.5 (1.7%)	2.4	0.7	0.7	**0.2**
$\text{Silos}_{1\cup2\cup3}$	0.0 (98.3%)	1.0 (0.0%)	1.8	1.0	0.9	**0.2**
$\mu \pm \sigma$	0.0 ± 0.0	0.35 ± 0.38	1.95 ± 0.47	0.78 ± 0.15	0.56 ± 0.26	$\mathbf{0.19 \pm 0.1}$

Table 1

Normalized tree edit distance for CFG-generated question and SQL pairs with accuracy reported in parentheses (average and std. dev. only applies to normalized tree edit distance). The exponential baseline sees all combinations of silos at training time, this is intractable and insecure, but has maximal performance. The generalization baseline sees all silos but not combinations of silos at training time, this is tractable but insecure. The other methods are used to build a SecureLLM. As described above, we do not include detailed reports on methods which underperform both LoraHub and PEM Addition. Note that our methods significantly outperform prior work. They retain all the generalization performance there is (since the generalization model sees all silos at once, while the fine-tunings each see silos separately, the generalization model should nominally perform better), even outperforming the generalization baseline.

of 128 tokens.

DeepSVDD generates its own anomaly score and we record this raw score for the comparison. We use the standard implementation of TF-IDF from the python package `sklearn`. Finally, we implement Compositional Perplexity as described above.

In fig. 2, we show anomaly score-density curves for $S1$ and HP, respectively from SecureSQL and CrossOverFanFic. The separation is immediately apparent for Compositional Perplexity; there is significantly more overlap with both DeepSVDD and TF-IDF, whereas LLM Perplexity shows much stronger separation and shorter tails minimizing overlap and maximizing the AUC.

Table 2 summarizes the graphs in fig. 2 by reporting only the AUC. From these tables, we can see that LLM Perplexity outperforms popular methods for Anomaly Detection. In addition to being the highest performing method, LLM Perplexity is also extremely lightweight because it relies on training a LoRA adapter for Llama-2-7b. In contrast, Deep-SVDD requires 10x more time to train. However, TF-IDF is the fastest and most lightweight method as it does not require any training. Nevertheless, accuracy is superior for LLM Perplexity when finetuned on Llama-2-7b.

SecureSQL Dataset

Inlier Data Silo	Deep SVDD	TF-IDF	LLM Perplexity
S1	92.0%	91.50%	**98.1%**
S2	92.9%	79.82%	**99.7%**
S3	95.3%	86.16%	**100.0%**

CrossOverFanFic Dataset

Inlier Data Silo	Deep SVDD	TF-IDF	LLM Perplexity
DCU	26.4%	75.15%	**100.0%**
HP	48.0%	63.69%	**100.0%**
MCU	15.5%	82.24%	**100.0%**

Table 2

Area Under the Curve (AUC) for the inlier data silo is shown for each anomaly detection method. Our proposed method of using Compositional Perplexity for anomaly detection outperforms published methods for a simple SQL dataset, and significantly outperforms published methods by a wide margin for a linguistically similar set of fan fiction stories.

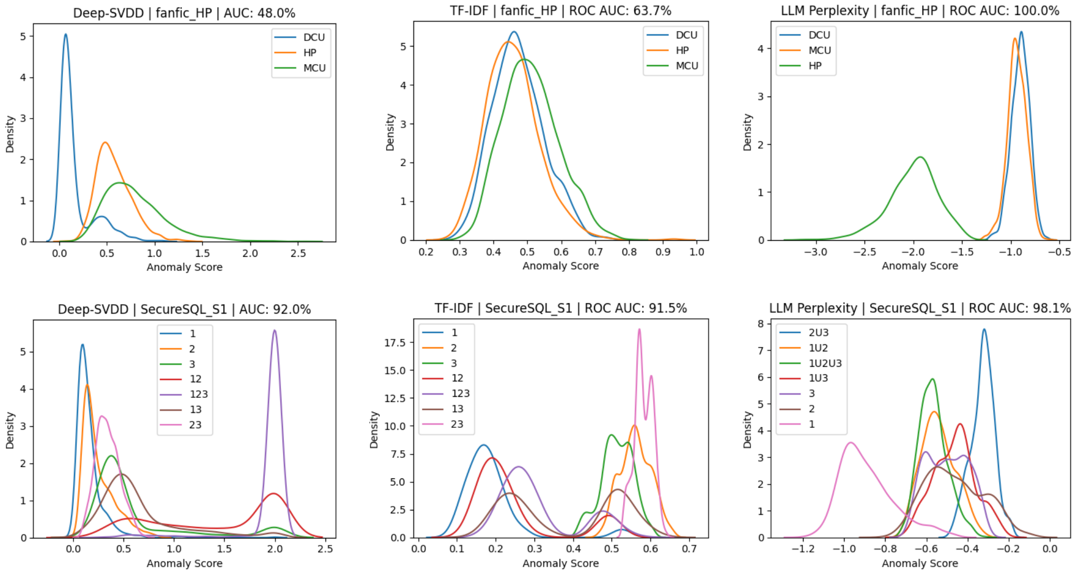

Figure 2: Anomaly Detection comparing Deep-SVDD, TF-IDF, and LLM Perplexity to detect when a leak has occured by comparing the outlier and inlier anomaly score. LLM Perplexity significantly outperforms common methods used for Anomaly Detection. Only Harry Potter Fanfic and S1 of the SQL dataset are shown for brevity. The ROC AUC scores for all data silos are in table 2; graphs for all data silos are shown in Appendix A. Note: anomaly scores are relative to each method and cannot be compared directly across methods.

Leak Identification Finally, we explore a new task that we call Leak Identification. This is essentially the refinement of anomaly detection (AD) where instead of detecting that a leak has occurred, we identify the most likely parameterized source of any given sample of text. Because AD methods like Deep-SVDD and TF-IDF are not designed for this task, we explore performance across a number of machine learning approaches based on research that shows that machine learning methods outperform other methods for DLP [9]. We showcase Leak Identification over the two datasets previously introduced, SecureSQL and CrossoverFanFic.

We implement SecureLLM using finetuned LoRA adapters from Llama-2-7b. The two datasets are divided into train, validation, and test splits. From the train set, we create the fine-tuned adapters. These fine-tuned adapters can be used in isolation to generate unsupervised predictions by comparing the generated loss for all possible compositions. We then normalize the seven unsupervised losses by subtracting the mean and dividing by the standard deviation of the data.

For supervised predictions comparable to other deep-learning methods, we train a Random Forest classifier from the validation set. Finally, using the the seven SecureLLM losses and the Random Forest classifier, we test on the test set to generate precision and recall scores, which are then used to report the final f1-score metric for data silo.

For comparison, we present LSTM, GRU, 1d-CNN, BiLSTM, and Transformer networks as benchmark methods along side our method using Llama-2-7b finetuned LoRA adapters. Shown in tables 3 and 4, our method is capable of near perfect Leak Identification for the SecureSQL dataset. In contrast, CrossoverFanFic appears to be a much more difficult task to accurately identify, particularly when the source samples comes from the crossover of three separate stories. This difference could be explained by the fact that the SecureSQL contains very unique key terms

Exp. 2						**Our Method**	**Our Method**
SecureSQL	LSTM	GRU	1d-CNN	BiLSTM	Trans.	**(Unsupervised)**	**Supervised**
Silos$_1$	0.61	0.65	0.87	0.52	0.26	0.59	**0.96**
Silos$_2$	0.61	0.73	0.83	0.58	0.46	0.91	**1.00**
Silos$_3$	0.66	0.93	0.92	0.88	0.61	0.67	**1.00**
Silos$_{1\cup2}$	0.47	0.46	0.87	0.43	0.54	0.82	**0.97**
Silos$_{1\cup3}$	0.46	0.62	0.80	0.37	0.33	0.55	**0.93**
Silos$_{2\cup3}$	0.61	0.74	0.80	0.53	0.48	0.50	**0.96**
Silos$_{1\cup2\cup3}$	0.38	0.65	0.87	0.52	0.46	0.21	**0.96**
Accuracy	0.54	0.68	0.85	0.55	0.46	0.60	**0.97**

Table 3

Leak Identification for SecureSQL dataset. We report the F1-Scores for each method according to the methodology described in Experiments: Supervised Leak Identification as well as the Weighted Average Accuracy for each method along the bottom line. All methods except Our Method (Unsupervised) are supervised methods. When comparing overall weighted accuracy it is notable that our Unsupervised method outperforms the supervised LSTM, BiLSTM, and Transformer. Our Supervised method clearly outperforms all other methods across all metrics. As previously mentioned, there is not a perfect method to compare against the unsupervised leak detection method, the F1-score is only provided to show how an unsupervised method outperforms even the best supervised methods in some instances.

that link to one specific data silo, whereas the CrossoverFanFic dataset is a lot more ambiguous over a small 128 token context window. However, despite these challenges identifying one of the datasets, our method doubled the best accuracy when compared to the other methods tested.

Exp. 2						**Our Method**	**Our Method**
X-overFanFic	LSTM	GRU	1d-CNN	BiLSTM	Trans.	**(Unsupervised)**	**Supervised**
HP	0.10	0.12	0.20	0.16	0.16	0.70	**0.99**
MCU	0.11	0.07	0.34	0.15	0.09	0.39	**0.98**
DCU	0.03	0.02	0.10	0.04	0.16	0.15	**0.98**
HP-MCU	0.59	0.57	0.59	0.58	0.59	0.00	**0.83**
HP-DCU	0.05	0.03	0.07	0.07	0.07	0.03	**0.20**
MCU-DCU	0.18	0.11	0.14	0.20	0.20	0.10	**0.64**
HP-MCU-DCU	0.03	**0.05**	**0.05**	0.02	0.02	0.00	**0.05**
Accuracy	0.33	0.31	0.36	0.35	0.35	0.14	**0.75**

Table 4

Leak Identification for CrossoverFanFic dataset. See table 3 and our methodology in Experiments: Supervised Leak Identification, for a detailed explanation. Our Supervised method clearly outperforms all other methods across all metrics. As previously mentioned, there is not a perfect method to compare against the unsupervised leak detection method, the F1-score is only provided to show how an unsupervised method outperforms even the best supervised methods in some instances

6. Conclusion

LLM security is critical in numerous commercial and government applications. We take a different view of LLM security compared to that of prior work, one where we import the traditional notion of access security to LLMs. This is enabled by the new compositional methods we introduce that prove themselves to be effective. We showed that these novel composition methods are able to take advantage of the generalization capabilities of the LLM with SQL edit distances that are the same or even at times better than the baseline LLM when it attempts to generalize. In other words, fine-tuning the LLM on each silo jointly, performs as well as fine-tuning on each silo individually and combining the fine-tunings. This is as much as one could hope for. Practical applications would need to use a far stronger underlying LLM to achieve high execution accuracy.

When leveraging inference-time composition, we blend concepts from the field of Data Loss Prevention (DLP) and Anomaly Detection (AD) to arrive at a novel interface. Compositional Perplexity is extremely effective and lightweight to detect the presence of anomalies in both generated text and human-written text. When combined with a Random Forest Classifier, it can also be used to identify and effectively categorize a single sample from a permuted compositional dataset.

There are numerous possible extensions of this work, including interfaces to document QA where each silo is a collection of documents rather than a database. One possible follow-up could look at the converse task: when given a question, determine the silos necessary to answer it. This could be used to monitor conversations or to automatically mark the appropriate access level of an exchange between users. Another possible direction would be to look at negative silos that exclude information. A negative silo would explicitly avoid a topic, which would prevent accidental leaks. Models could rewrite text or data to refer to or exclude particular silos. The traditional world of access security is rich with problems for LLMs to address, and our work opens up the path to doing so. In addition, by providing provable security, i.e., there can be no leaks from silos the user doesn't have access to, we take a key step toward enabling the use of LLMs in secure environments.

7. Acknowledgements

This work was supported by the Center for Brains, Minds, and Machines, NSF STC award CCF-1231216, the NSF award 2124052, the MIT CSAIL Machine Learning Applications Initiative, the MIT-IBM Watson AI Lab, the CBMM-Siemens Graduate Fellowship, the DARPA Mathematics for the DIscovery of ALgorithms and Architectures (DIAL) program, the DARPA Knowledge Management at Scale and Speed (KMASS) program, the DARPA Machine Common Sense (MCS) program, the Air Force Office of Scientific Research (AFOSR) under award number FA9550-21-1-0399, the United States Air Force Research Laboratory and the Department of the Air Force Artificial Intelligence Accelerator under Cooperative Agreement Number FA8750-19-2-1000, and the Developing the Airmen We Need–Education (DAWN-ED) PhD fellowship.

8. Disclaimer

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force, the United States Space Force, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Christian Arnold is an active duty officer of the United States Space Force on a Department of the Air Force funded fellowship.

References

[1] N. Mangaokar, A. Hooda, J. Choi, S. Chandrashekaran, K. Fawaz, S. Jha, A. Prakash, Prp: Propagating universal perturbations to attack large language model guard-rails, arXiv:2402.15911 (2024).

[2] S. Banerjee, S. Layek, R. Hazra, A. Mukherjee, How (un) ethical are instruction-centric responses of llms? unveiling the vulnerabilities of safety guardrails to harmful queries, arXiv:2402.15302 (2024).

[3] A. Dutta, A. Khorramrouz, S. Dutta, A. R. KhudaBukhsh, Down the toxicity rabbit hole: A framework to bias audit large language models with key emphasis on racism, antisemitism, and misogyny, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence AI for Good, 2024, pp. 7242–50.

[4] M. Andriushchenko, F. Croce, N. Flammarion, Jailbreaking leading safety-aligned llms with simple adaptive attacks, arXiv:2404.02151 (2024).

[5] C. Huang, Q. Liu, B. Y. Lin, T. Pang, C. Du, M. Lin, Lorahub: Efficient cross-task generalization via dynamic lora composition, arXiv:2307.13269 (2023).

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv:2106.09685 (2021).

[7] J. Zhang, S. Chen, J. Liu, J. He, Composing parameter-efficient modules with arithmetic operations, arXiv:2306.14870 (2023).

[8] J. Guo, Y. Li, M. Wang, X. Qiao, Y. Wang, H. Shang, C. Su, Y. Chen, M. Zhang, S. Tao, H. Yang, Y. Qin, The HW-TSC's speech to speech translation system for IWSLT 2022 evaluation, in: International Conference on Spoken Language Translation, 2022, pp. 293–297.

[9] A. Guha, D. Samanta, A. Banerjee, D. Agarwal, A deep learning model for information loss prevention from multi-page digital documents, IEEE Access 9 (2021) 80451–80465.

[10] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (2009) 1–58.

[11] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural computation 13 (2001) 1443–1471.

[12] M. Kim, J. Kim, J. Yu, J. K. Choi, Unsupervised deep one-class classification with adaptive threshold based on training dynamics, in: 2022 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2022, pp. 39–46.

[13] A. Chronopoulou, M. E. Peters, A. Fraser, J. Dodge, Adaptersoup: Weight averaging to improve generalization of pretrained language models, arXiv:2302.07027 (2023).

[14] Y. Du, S. Li, I. Mordatch, Compositional visual generation and inference with energy based models, arXiv:2004.06030 (2020).

[15] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, Peft: State-of-the-art parameter-efficient fine-tuning methods, https://github.com/huggingface/peft, 2022.

[16] K. Zhang, J. T. Wang, D. Shasha, On the editing distance between undirected acyclic graphs, International Journal of Foundations of Computer Science 7 (1996) 43–57.