



25. Heimspiel

**”AI for the rest of us” ...
... aber sicher & mit Sicherheit**

Udo Schneider

TrendAI

Udo_Schneider@trendmicro.com



25. Heimspiel

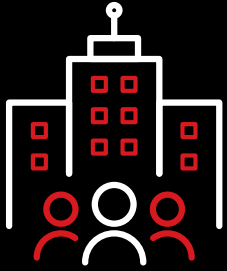
Vom Hype zur Realität

Wie vor allem etablierte Unternehmen mit künstlicher Intelligenz echten Mehrwert schaffen

Udo Schneider

TrendAI

Udo_Schneider@trendmicro.com



KI verändert unsere Welt...

... und transformiert
Leben, Arbeit & Freizeit







16.07.1945



30.11.2022



CHATGPT

30.11.2022



Stille Post



Article
AI models collapse when trained on recursively generated data

Article
AI models collapse when trained on recursively generated data
https://doi.org/10.1101/2024.05.20.592404

<https://doi.org/10.1038/s41598-024-62760-0>
 Received: 20 December 2023
 Accepted: 16 July 2024
 Published online: 26 July 2024
 © The Author(s) 2024

being discrete, making the Markov chain over the parameters of the data functions, we will get in the model parameterization allows for the only possible absorbing states are the delta functions. On the basis of the discussion above, we can see how both early model collapse and only the low probability events get cut off in the basis of the basis of the basis, in which the process begins to collapse into a single model, which arise in the case of discrete distributions with perfect functions.

Theorem 1.1 (Gaussian model collapse). Assume that $\mathbf{z}$ is sampled from a distribution $p(\mathbf{z})$ that is not Gaussian. Then, as $n \rightarrow \infty$, the distribution of $\mathbf{z}^n$ converges to a Gaussian distribution.

Theorem 1.1 (Gaussian model collapse). Assume the original data are sampled from distribution $\mathcal{D}$, (not necessarily Gaussian), with non-zero sample variance. Assume $\mathbf{x}^*$ are fit recursively using the unbiased sample mean and variance estimators from the previous generation, $\{\hat{\mu}_t, \hat{\Sigma}_t = \hat{\Sigma}(\mathbf{x}_t^*)\}_{t=1}^T$, with a fixed sample size. Then,

which $\hat{w}_t$ denotes the Wasserstein distance between the true distribution and its approximation at generation t . In two words, this implies that at generation t , the true distribution diverges arbitrarily far from the t th generation approximation. The results are very analogous: if θ also collapses in this theorem illustrating the effect of late stage model collapse, the process begins to collapse to be a zero variance, in which collapse can also be seen and the interested reader is referred to supplementary Materials for a more in-depth discussion.

Model collapse in language models

and GMMs—in the Supplementary Material available in ref. 13.

Model collapse is universal across various families of machine learning models. Yet, if small models such as GMMs and VAEs are trained from scratch, LLMs are different. They are so expensive to train from scratch that they are typically initialized with pre-trained text corpora.

[illegible][illegible]

of a non-zero probability that information can get lost at every step of resampling.

- **Functional expressivity error.** This is a secondary type of error arising owing to limited function approximator expressivity. In particular, neural networks are only universal approximators in the limit of infinity. As a result, a neural network can introduce a non-zero likelihood inside the support of the original distribution or provide a likelihood inside the support of the original distribution. A classic example of the expressivity error is if we tried fitting a mixture of Gaussians with a single Gaussian. Even if we have perfect information about the data distribution that is, information from all samples, model errors will be introduced because of the absence of the other two types of error, this can only occur at the first generation.
- **Functional approximation error.** This is a secondary type of error arising owing to limited function approximator expressivity. In particular, neural networks are only universal approximators in the limit of infinity. As a result, a neural network can introduce a non-zero likelihood inside the support of the original distribution or provide a likelihood inside the support of the original distribution. A classic example of the expressivity error is if we tried fitting a mixture of Gaussians with a single Gaussian. Even if we have perfect information about the data distribution that is, information from all samples, model errors will be introduced because of the absence of the other two types of error, this can only occur at the first generation.

- **Functional approximation error.** This is a secondary type of error arising primarily from the limitations of learning procedures, for

example, the structural bias of stochastic gradient descent is a source of systematic error in the approximation. This error can be viewed as an approximation of the true distribution, but each of the above can cause model collapse. The approximation power can even be a disadvantage. More often than not, we get a cascading effect, where the approximation of the true distribution and inaccuracies combine to cause a cascading effect, overfitting the density model causes the model to assign high-density regions to low-frequency in the training set support, these low-frequency regions are then assigned to high-frequency. It is worth noting that, for example, computers have limited precision, so mathematical intuition has limited predictive power.

[illegible]

...general owing to floating-point representations



“The mystery of em-dashes”

- Neuere Modelle wurden mit anderen Daten trainiert, die mehr Gedankenstriche enthalten
- Gedankenstriche sparen Token

What came out

Both hypotheses held up.

H1: The drastic increase in em-dash usage for newer models

To confirm **H1**, we need to prove that the older model generated less em-dashes. The increases of em-dashes is dramatic. Later models showed 10-fold increase. We can now speculate that the reason for this is the digitalization of older books.

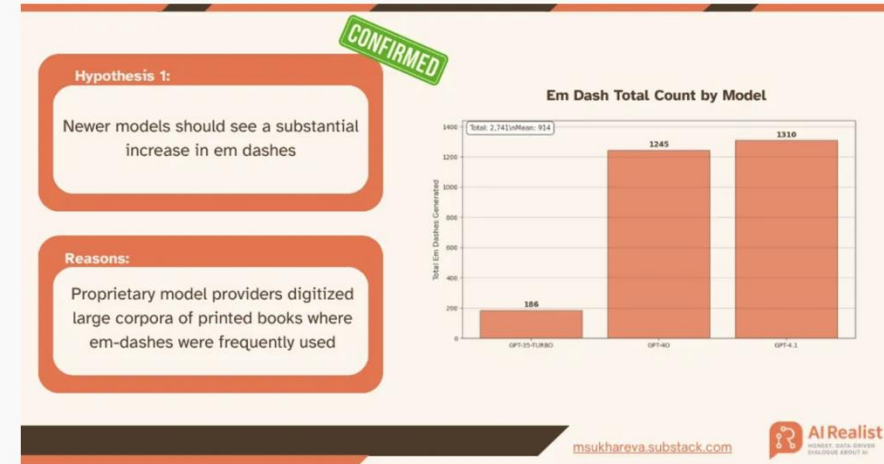


Table 2: H1 is confirmed. Later models generate more than 10x more em-dashes per story.

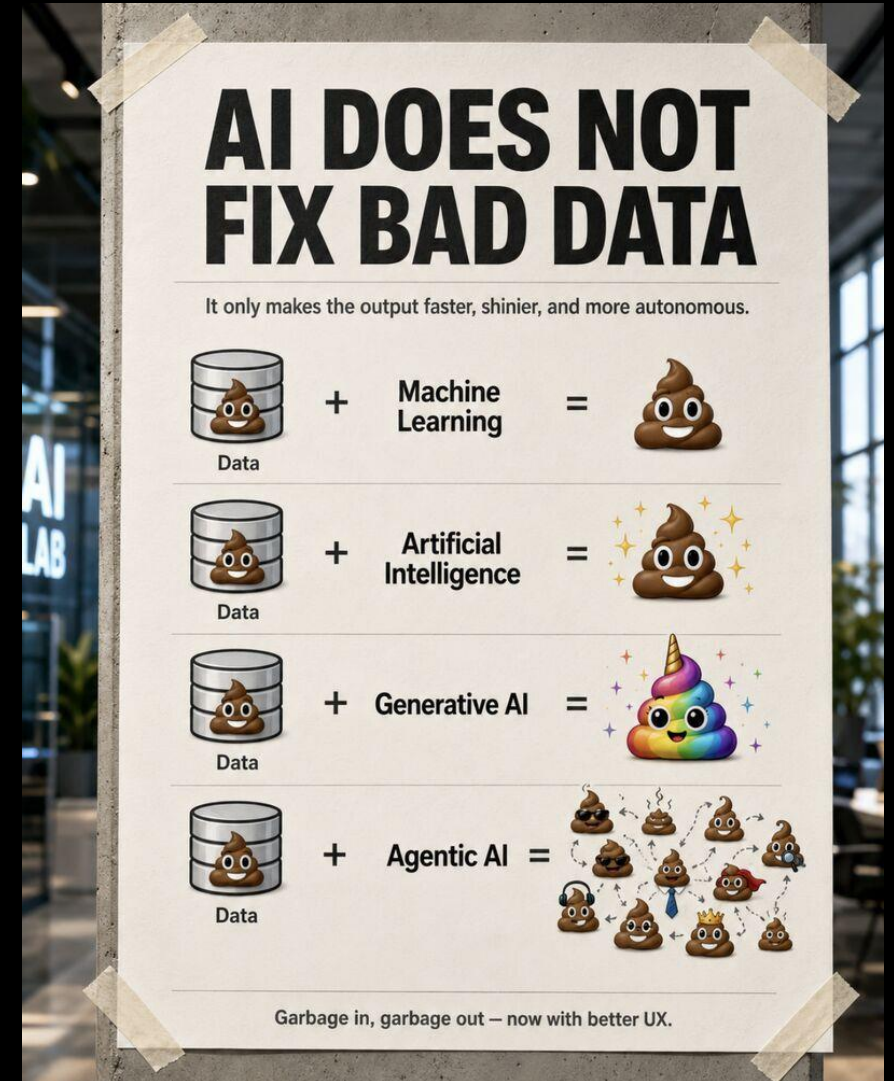


Indeed, the value of data collected about genuine human interactions with systems will be increasingly valuable in the presence of LLM-generated content in data crawled from the Internet.

Shumailov, I. et al. AI models collapse when trained on recursively generated data. Nature 631, 755–759 (2024).

Datenqualität als Fundament

- „Garbage in – Garbage out“
 - KI ist nur so gut wie die Datenbasis
 - Strukturierte, saubere & *nachvollziehbare* Daten sind entscheidend
- Daten strukturieren
 - Schema
 - OCR



Text Daten strukturieren

Patient John Smith is prescribed Lisinopril 20 mg once daily for hypertension, Metformin 500 mg twice daily for type 2 diabetes, and Atorvastatin 40 mg once daily for hyperlipidemia. Patient Maria Lopez is prescribed Sertraline 50 mg once daily for depression and Albuterol inhaler, two puffs every 6 hours as needed for asthma. Patient Ravi Patel is prescribed Amoxicillin 500 mg three times daily for 7 days for a bacterial sinus infection, and Ibuprofen 400 mg every 8 hours as needed for pain.

Schema

```
- name: John Smith
  prescriptions:
    - drug_name: Lisinopril
      dosage: 20 mg once daily
    - drug_name: Metformin
      dosage: 500 mg twice daily
    - drug_name: Atorvastatin
      dosage: 40 mg once daily
- name: Maria Lopez
  prescriptions:
    - drug_name: Sertraline
      dosage: 50 mg once daily
    - drug_name: Albuterol inhaler
      dosage: two puffs every 6 hours as needed
- name: Ravi Patel
  prescriptions:
    - drug_name: Amoxicillin
      dosage: 500 mg three times daily for 7 days
    - drug_name: Ibuprofen
      dosage: 400 mg every 8 hours as needed
```

Bild Daten strukturieren

Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion

Nikolaos Livathinos, Christoph Auer*, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Panagiotis Vagenas, Cesar Berrosio, Matteo Omenetti, Kasper Dinkla, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, Peter W. J. Staar

ktor Kuropiatnyk, Peter W. J. Staar

IBM Research, Rüschlikon, Switzerland

Please send correspondence to: deepsearch-core@zurich.ibm.com

Abstract

We introduce *Docling*, an easy-to-use, self-contained, MIT-licensed, open-source toolkit for document conversion, that can parse several types of popular document formats into a unified, richly structured representation. It is powered by state-of-the-art machine learning models for document layout (DocLayoutNet) and table structure recognition (TableFormer), and runs efficiently on commodity hardware in a small resource budget. Docling is released as a Python package and can be used as a Python API or as a CLI tool. Docling's modular architecture and extensible document representation make it easy to implement extensions, new models, and customizations. Docling has been already integrated in other popular open-source frameworks (e.g., LangChain, LlamaIndex, spaCy), making it a natural fit for the processing of documents and the development of high-end applications. The open-source project has already gathered 10k stars on GitHub and more than a month and was reported as the No. 1 trending repository in GitHub worldwide in November 2024.

Repository — <https://github.com/DS4SD/docling>

1 Introduction

Converting documents back into a unified, machine-processable format has been a major challenge for decades due to their huge variability in formats, weak standardization of printing, optimization of the layout for the human eye, structural elements and metadata. With the advent of LLMs and popular application patterns such as retrieval-augmented generation (RAG), leveraging the rich content embedded in PDF documents has become a top priority. PDF documents have become ever more relevant. In the past decade, several powerful document understanding solutions have emerged on the market, most of which are commercial software, SaaS offerings for hyperscalers (Auer et al., 2022), and more recently, open-source solutions (Liu et al., 2022). However, they incur a cost (e.g., for licensing or LLM inference) and cannot be run easily on local hardware. Meanwhile, only a handful of different open-source tools cover PDF, MS Word, MS PowerPoint, etc. (see Table 1). They lack the ability to extract feature and quality gap to proprietary solutions.

These authors contributed equally.
Copyright © 2023, Association for the Advancement of Artificial
Intelligence (www.aaai.org). All rights reserved.

With *Docling*, we recently open-sourced a very capable and efficient document conversion tool which builds on the powerful, specialized AI models and datasets for layout analysis and table extraction that we have previously developed and presented in the recent past (Li/Vivathanos et al. 2021; Pitzmann et al. 2022; Lysak et al. 2023). *Docling* is designed as a simple, self-contained Python library with permissive MIT license and a modular architecture that allows for easy extensibility and addition of new features and models. Since its launch in July 2024, *Docling* has attracted considerable attention in the AI developer community and ranks top 10 on GitHub's monthly trending page for open-source projects in the AI category of writing. On October 16, 2024, *Docling* reached a major milestone with version 2, introducing several new features and concepts, which we outline in this updated technical report. We also provide detailed performance evaluation on various benchmarks, and comparisons to other open-source tools.

The following list summarizes the features currently available on Docling:

- Parses common document formats (PDF, Images, MS Office formats, HTML) and exports to Markdown, JSON, and HTML.
- Applies advanced AI for document understanding, including detailed page layout, OCR, reading order, figure extraction, and table structure recognition.
- Establishes a unified `DoclingDocument` data model for rich document representation and operations.
- Provides fully local execution capabilities making it suitable for sensitive data and air-gapped environments.
- Has an ecosystem of plug-and-play integrations with prominent generative AI development frameworks, including LangChain and LlamaIndex.
- Can leverage hardware accelerators such as GPUs.

2 State of the Art

Document conversion is a well-established field with numerous solutions already available on the market. These solutions can be categorized along several key dimensions, including open vs. closed source, permissive vs. restrictive licensing, Web APIs vs. local code deployment, susceptibility

Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion

Nikolaos Livathinos [§](-), Christoph Auer [§](-), Matsys Lysak, Ahmed Nassar, Michele Dolfi, Panagiotis Vagenas, Cesar Berrospi, Matteo Omenetti, Kasper Dinkla, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, Peter W. J. Staar

IBM Research, Rüschlikon, Switzerland

Please send correspondence to: deepsearch-core@zurich.ibm.com

Abstract

We introduce Docling, an easy-to-use, self-contained, MITlicensed, open-source toolkit for document conversion, that can parse several types of popular document formats into a unified, richly structured representation. It is powered by state-of-the-art specialized AI models for layout analysis (DocLayNet) and table structure recognition (TableFormer), and runs efficiently on commodity hardware in a small resource budget. Docling is released as a Python package and can be used as a Python API or as a CLI tool. Docling's modular architecture and efficient document representation make it easy to implement extensions, new features, models, and customizations. Docling has been adopted by several major open-source frameworks (e.g., LangChain, LlamaIndex, and LlamaParse) as it is natural fit for the growth of documents and the development of high-end applications. The open-source community has fully engaged in using, promoting, and developing for Docling, which gained 10k stars on GitHub in less than a month and was reported as the No. 1 trending repository in GitHub worldwide in November 2024.

Repository - <https://github.com/DS4SD/docling>

1 Introduction

Converting documents back into a unified machine-readable format has been a major challenge for decades due to their huge variability in formats, weak standardization and printing-unrelated characteristics, which often discards structural features and metadata. With the advent of LLMs and popular application patterns such as retrieval-augmented generation (RAG), leveraging the rich content embedded in PDFs, Office documents, and scanned document images has become ever more relevant. In the past decade, several powerful document understanding solutions have emerged on the market, most of which are commercial, such as Saas offerings on hyperscalers (Auer et al. 2022) and most recently, multimodal vision-language models. Typically, they incur a cost (e.g., for licensing or LLM inference) and cannot be run easily on local hardware. Meanwhile, only a handful of different open-source tools cover PDF, MS Word, MS PowerPoint, Images, or HTML conversion, leaving a significant feature and quality gap to proprietary solutions.

These authors contributed equally.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

With Docling, we recently open-sourced a very capable and efficient document conversion tool which builds on the powerful, specialized AI models and datasets for layout analysis and table structure recognition that we developed and presented in the recent past (Livathinos et al. 2021; Pfizmann et al. 2022; Lysak et al. 2023). Docling is designed as a simple, self-contained Python library with permissive MIT license, running entirely locally on commodity hardware. Its code architecture allows for easy extensibility and addition of new features and models. Since its launch in July 2024, Docling has attracted considerable attention in the AI developer community and ranks top on GitHub's monthly trending repositories with more than 10,000 stars at the time of writing. On October 16, 2024, Docling reached a major milestone with version 2, introducing several new features and concepts, which we outline in this updated technical report, along with details on its architecture, conversion speed benchmarks, and comparisons to other open-source assets.

The following list summarizes the features currently available on Docling:

- Parses common document formats (PDF, Images, MS Office formats, HTML) and exports to Markdown, JSON, and HTML.
- Applies advanced AI for document understanding, including detailed page layout, OCR, reading order, figure extraction, and table structure recognition.
- Establishes a unified DoclingDocument data model for rich document representation and operations.
- Provides fully local execution capabilities making it suitable for sensitive data and air-gapped environments.
- Has an ecosystem of plug-in plug integrations with prominent generative AI development frameworks, including LangChain and Llamaindex.
- Can leverage hardware accelerators such as GPUs.

2 State of the Art

Document conversion is a well-established field with numerous solutions already available on the market. These solutions can be categorized along several key dimensions, including open vs. closed source, permissive vs. restrictive licensing, Web APIs vs. local code deployment, susceptibility

A stylized, glowing orange wireframe lighthouse stands on a dark blue background. The lighthouse is composed of a grid of lines, with a small arched entrance at the base. A bright orange beam of light emanates from the top of the lighthouse, extending across the upper portion of the image. The ground is represented by a perspective grid of light blue lines, and the background features a range of stylized, jagged mountains in the distance. The overall aesthetic is digital and futuristic.

Leuchtturmprojekte

95%

of generative AI pilots
at companies are
failing¹

88%

of AI pilots fail to
reach production²

40%

of Agentic AI projects
will be canceled by
end of 2027³

¹ <https://fortune.com/2025/08/18/mit-report-95-percent-generative-ai-pilots-at-companies-failing-cfo/>

² <https://www.cio.com/article/3850763/88-of-ai-pilots-fail-to-reach-production-but-thats-not-all-on-it.html>

³ <https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027>

Erfolgsfaktor Mitarbeitende

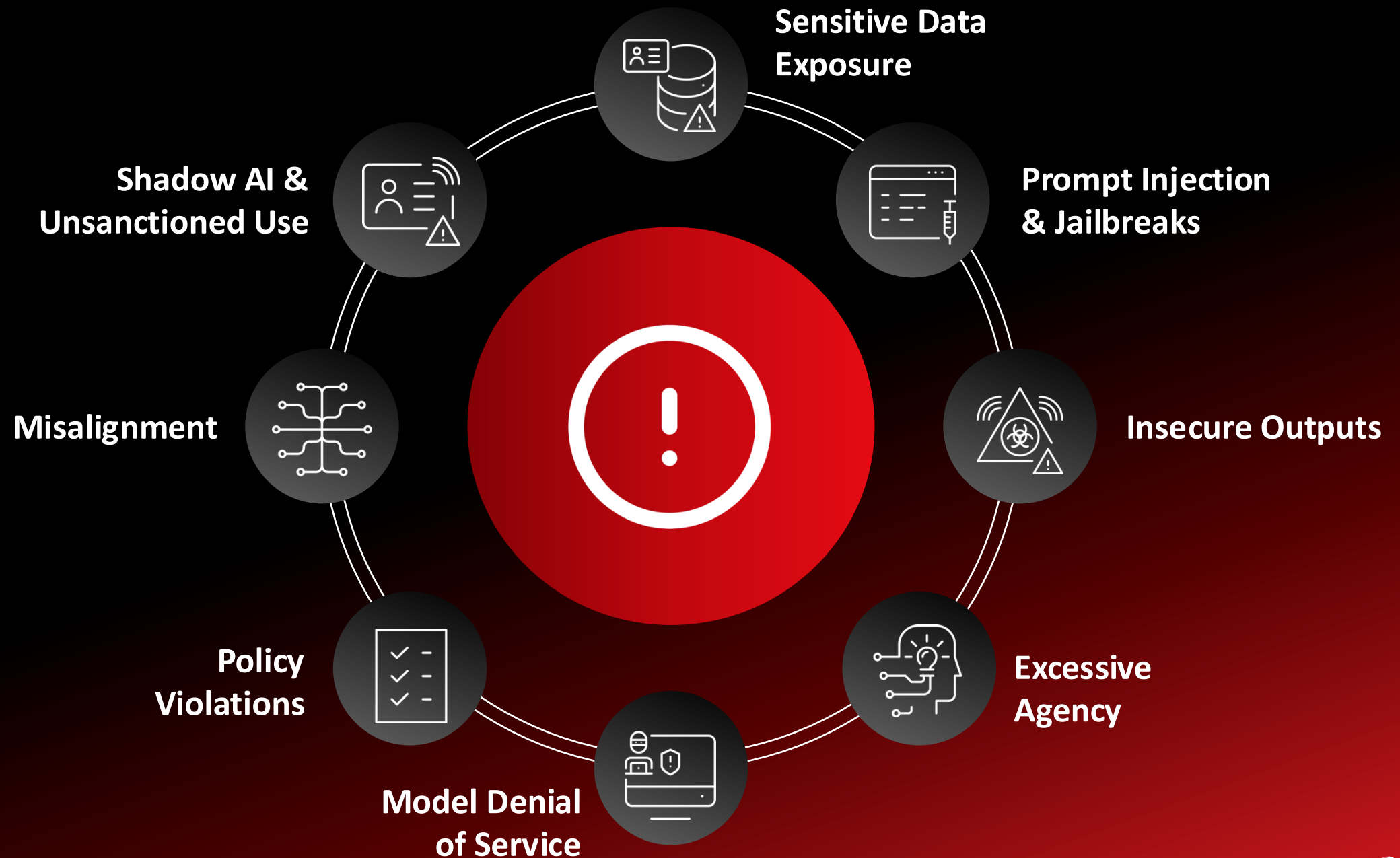
- Einbeziehung der Mitarbeiter von Anfang an!
- Sie kennen Schwachstellen und Routineaufgaben
- Sie liefern praxisnahe, realistische KI-Ideen

**Bottom-up-Projektideen,
die wirklich relevant sind!**

KI-Kompetenz aufbauen

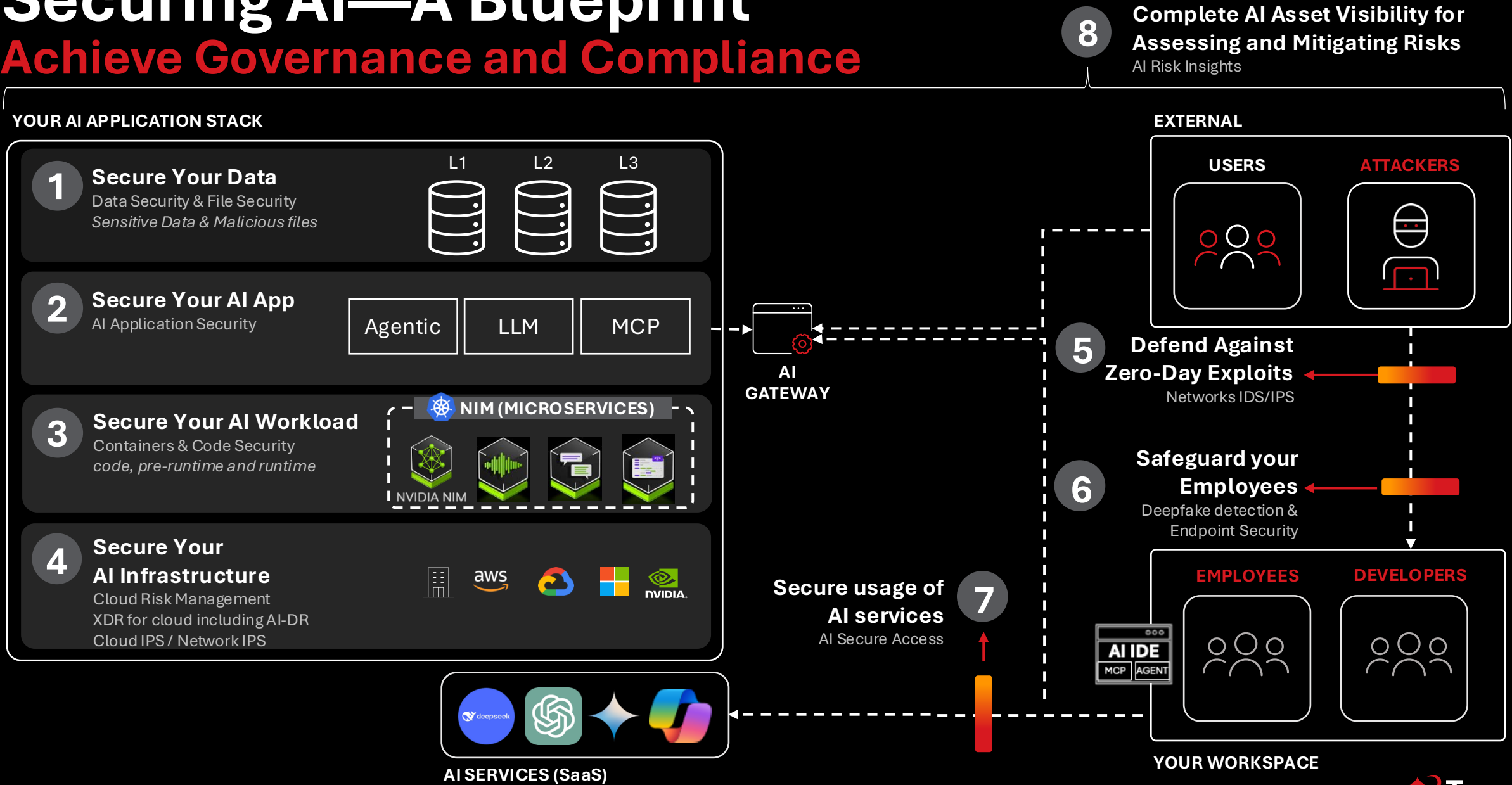
- Schulungen für alle Mitarbeitende
- Verständnis fördern über:
 - Grundlagen von KI & maschinellem Lernen
 - AI/ML Roads less travelled (Structure, Embeddings, CV)
 - Grenzen und Risiken (Bias, Halluzinationen, Datenschutz)

Erst KI-Kompetenz...
... dann Projekte!



Securing AI—A Blueprint

Achieve Governance and Compliance



OWASP Top 10

LLM01:2025 Prompt Injection

LLM02:2025 Sensitive Information Disclosure

LLM03:2025 Supply Chain

LLM04:2025 Data and Model Poisoning

LLM05:2025 Improper Output Handling

LLM06:2025 Excessive Agency

LLM07:2025 System Prompt Leakage

LLM08:2025 Vector and Embedding Weakness

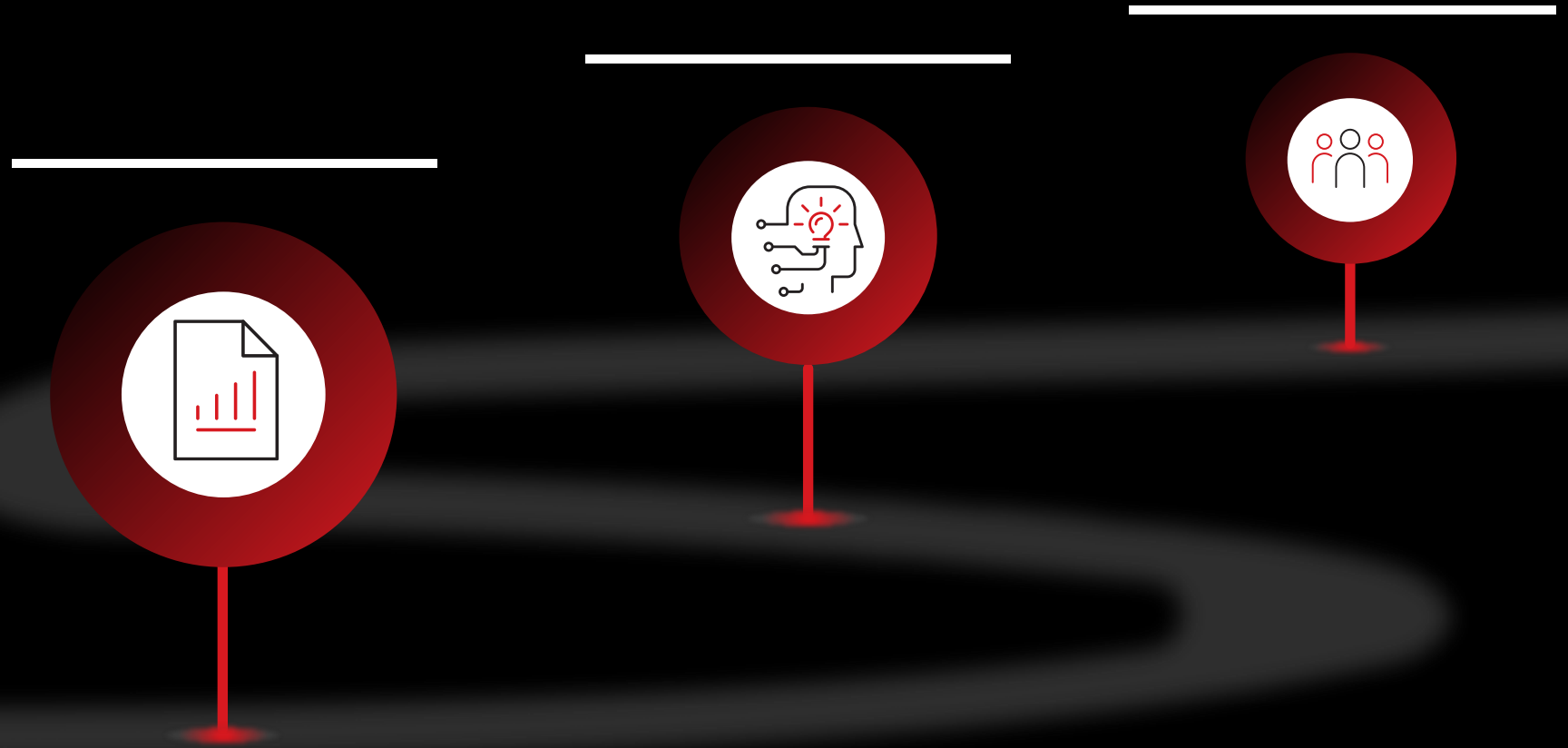
LLM09:2025 Misinformation

LLM10:2025 Unbounded Consumption

TrendAI™ AI Security Assessment



Zusammenfassung





AI Fearlessly



Vielen Dank!

Udo Schneider

Udo_Schneider@trendmicro.com

