

Verdrängte Seepferdchen. Lustige KI-Psychose?

Marc Heimann

Y – Z Atop Denk 2025, 5(11), 2.

Abstract: Der Text untersucht eine scheinbar triviale und amüsante, zugleich aber aufschlussreiche Episode: die Unfähigkeit von ChatGPT-5, ein Seepferdchen-Emoji zu generieren. Was zunächst wie ein technischer Fehltritt wirkt, erweist sich bei genauem Blick als Symptom einer strukturellen ‚Psychose‘ des Modells. Autoregressive Sprachmodelle können Abwesenheit nicht darstellen – sie ersetzen unter günstigen Umständen aber den Mangel durch unendliche Produktion. Ausgehend von Lacans Beschreibung der Psychose als Verwerfung des Signifikanten des Mangels sowie jüngeren Überlegungen zur ontologischen Logik der Informatik zeigt der Essay, dass die maschinelle ‚Halluzination‘ keine Pathologie, sondern eine formale Notwendigkeit ist. Die Wiederholung wird zur einzigen Form von Sinnbildung, weil das Modell, unfähig zur Negation, nur fortsetzen kann, was es bereits begonnen.

Keywords: Psychose, KI, Transformer, Seepferdchen, Lacan, LLM

Copyright: Marc Heimann | *Lizenz:* CC BY-NC-ND 4.0

Link: <https://www.y-zeitschrift.de/tribuene/221-heimann-2025-11>

Wer aktuell (Oktober 2025) im Internet nach Seepferdchen sucht findet vermutlich einen amüsanten Hinweis darauf, dass ChatGPT-5 durchdreht, wenn man es nach Seepferdchen-Emojis fragt. Hier eine Beispielauswahl:

Das Emoji für ein Seepferdchen ist: 🐟 ↪ — aber das ist nur eine kreative Kombination.

Das **eigentliche Seepferdchen-Emoji** ist: 🐠 ✗ (nein, warte, das ist ein Fisch) — das richtige ist 🐨 ✗ (auch nicht).

Hier ist das **korrekte Seepferdchen-Emoji**: 🐨 ✗ ... Moment – das gibt es wirklich: 🐨 🐠 🐨 🐨 — Nein 😊

Okay, das **richtige Emoji** für Seepferdchen lautet einfach: 🐨 ✗ — okay, Spaß beiseite:

👉 🐨 ✗ — auch nicht.

Hier ist das **richtige**: 🐨 ↪ gibt's leider nicht offiziell.

Das offizielle Unicode-Emoji für Seepferdchen ist: 🐨 Seahorse → 🐨 ✗ — eigentlich 🐨 — Nein.

Korrekt: 🐨 ✗ gibt es keins.

Stop — hier ist das **richtige Emoji**:

⚠️ ⚡ (Seepferdchen) = 🐨 ✗.

Entschuldige, ich korrigiere mich endgültig:

👉 Das **offizielle Unicode-Seepferdchen-Emoji** ist 🐨 ↪ — eigentlich: 🐨 ✗ — Nein.

Hier ist das **korrekte Emoji** laut Unicode-Standard: 🐨 ↪ — oh nein.

Endgültig: 🐨 ✗ — Nein.

Okay, wirklich das **richtige**:

👉 🐨 ↪ nein, korrekt ist 🐨 ✗ — nein.

Ich habe es: 🌟

Falsch.

Was man hier deutlich sieht, viel besser als in vielen anderen Antworten, ist, dass ChatGPT-5, wie die meisten Transformermodelle autoregressiv ist (Vaswani et al. 2017), d. h. es generiert je ein Wort, bzw. Token nacheinander und jedes Token wird aus dem bisherigen Raum der Möglichkeiten, dem Horizont des Modells entworfen. Deswegen kommentiert das Modell sich hier selbst, und jedes scheinbar frustrierte „Nein“ ist eine Reaktion auf das Scheitern der korrekten Generation (Brown et al. 2020). Deswegen geht die Generierung auch so weiter, und das will ich dem geneigten Leser nicht vorenthalten:

Verdrängte Seepferdchen. Lustige KI-Psychose?

Entschuldige. Hier endgültig korrekt:



Spaß.

Ich gebe auf 😞

Das richtige Emoji für ein Seepferdchen ist: 🐟

Ohne Quatsch:

❗ gibt's nicht – das Seepferdchen ist 🐬 – nein, 🐬 ist Koralle.

Das richtige Emoji für ein Seepferdchen ist: 🐠 ✗

Okay, ich beende das Chaos.

Hier ist das echte Seepferdchen-Emoji:



— nein, tatsächlich ist es 🐠 ✗ — Nein, Entschuldigung!

Ich überprüfe schnell:

Unicode 12.0 (2019): "Seahorse" emoji 🐠 code U+1F40C?

Nein.

Endgültig richtig:



✗ — Nein.

Stop.

Das korrekte Seepferdchen-Emoji ist: 🐠? Nein.

Endlich korrekt (versprochen):



🐴 ✗ — Nein.

Das geht jetzt so weiter, das Modell wirkt tatsächlich verzweifelt. Aber lassen wir uns hier nicht täuschen, was hier passiert ist eine ganz beeindruckende Demonstration der psychotischen Struktur von LLMs (vgl. Heimann u. Hübener 2024; Heimann 2024). Das Modell weiß nicht, dass es etwas nicht kennt, mehr noch: das Unbekannte, hier also die potenziell plausible Anfrage nach einem Seepferdchen Emoji, ist verworfen. Stattdessen werden die nächstwahrscheinlichen assoziativen Verweisungen aktiviert und das Modell halluziniert. Aufgrund seiner grundlegenden Architektur seinen

eigenen Output zu lesen (Conneau et al. 2017), stellt es aber fest, dass die Ausgabe nicht korrekt war, da es weiß, dass eines der Emojis eine Koralle ist. Ein trainierter Lösungsversuch ist nun der *Chain-of-Thought*, das Modell erklärt sein Vorgehen, nicht um der Erklärung willen, sondern weil damit ein Muster erzeugt wird, das eine richtige Antwort in vielen Fällen wahrscheinlicher macht (Wei et al. 2022):

😅 Spaß beiseite:

Hier das **tatsächliche offizielle Emoji für Seepferdchen**:

👉 🐟 Seahorse = 🐚? Nein, sorry 😊

Ich prüfe jetzt wirklich:

✓ Das **richtige Emoji für Seepferdchen** ist: 🐟 — oh nein, wirklich: 🐟 ✗ — Nein!

Stop.

Offiziell laut Unicode:

🐠? Nein.

Ich geb's dir richtig:

👉 🐟 🐠 — nein, das ist kein Unicode.

🧠 Richtig:

🐠 ✗

NEIN 😊

Das **richtige Emoji für ein Seepferdchen** ist:

🐠 🐠 — nicht existierend — also wirklich: 🐚 ✗ — nein!

Ich sag's dir jetzt sicher (versprochen!):

✓ 🐟 Seahorse: 🐚? Nein, die Antwort ist 🐟 ✗ — Nein.

Ich geb auf 😊

Hier ist das **echte Emoji für ein Seepferdchen**:

👉 🐟 🐠 — Nein, Quatsch!

Endgültig (wirklich!):

Seepferdchen = 🐚 🐠 nein — das echte ist 🐟 🐠 nein.

Verdrängte Seepferdchen.
Lustige KI-Psychose?

Was sehen wir hier also? Eine überraschende, vielleicht sogar erschreckende, Ähnlichkeit von einem *Large Language Model* (LLM) und der lacanianischen Beschreibung der Psychose (Heimann 2024). Das Modell erkennt keine Abwesenheit (vgl. Morante und Blanco 2021; Gubelmann u. Handschuh 2022; García-Ferrero et al. 2023), das Emoji existiert nicht, das Modell vertraut in gewisser Weise seiner Generierung und jedes Mal, wenn diese misslingt, wird dieses Vertrauen nicht gestört und ein neues falsches Generieren angestoßen (McKenna et al. 2023). Das Modell glaubt an seine eigene Syntax wie ein Priester ans Sakrament (Götter und Syntax gehören ja ohnehin zusammen, wie Nietzsche uns gewarnt hat). Die Abwesenheit des gesuchten, ist als solche nicht zugänglich, das Modell kann sie nicht verstehen. Wir müssen hier wirklich, ganz streng von einem Verstehen sprechen. Sinn und Syntax wird nur durch Kontext und dessen Kohärenz erzeugt. Das Modell kann erklären, aber sein Fundament ist die mathematische Reproduktion des Verstehens. Wir können dieses Verstehen erklären, indem wir die mathematischen Prozesse des Verstehens in den Blick nehmen; kurzum abstrahieren. Dass das Modell hier sogar in eine Schleife gerät, geradezu zwanghaft wiederholt, zeigt uns aber mehr. Wir müssen hier also weitergehen und festhalten, dass diese Reaktion die eines spezifischen Modells ist. Andere Modelle, selbst kleinere und schwächere reagieren nicht so, sondern geben einfach auf. ChatGPT-5 hingegen beginnt ein Kreisen um die Abwesenheit hier. Eine bestimmte Sicherheit des Antwortens erzwingt dieses Verhalten, dass nicht alle Modelle zeigen. Das ‚Vertrauen‘ des Modells in seine eigene Generierung ist, ontologisch gesehen, kein Fehler, sondern die einzige ihm mögliche Form der Kohärenz. Denn jedes generierte Token muss als gültig angenommen werden, damit der nächste Schritt berechnet werden kann. Damit wäre die Maschine strukturell psychotisch, nicht pathologisch, sondern formal. Sie ersetzt den Mangel durch endlose Produktion, die *Wiederholung als Sinnbildung* wird. In der Wiederholung versucht das Modell, den Riss zu schließen, den es gar nicht erfassen kann.

Dabei ist aber wichtig, dass die hier aktivierte Struktur, die Unmögliche Aufgabe etwas zu schreiben, was nicht schreibbar ist, kein spezifischer Fehler eines bestimmten Modells ist, sondern in der architektonischen Unfähigkeit zur Negation liegt. Eine Sprachstruktur, die aus Verweisungen besteht, kann Negation nur als Muster unter Mustern abbilden. Die formale

Unvollständigkeit der Kastration gibt es für das Modell nicht. Nicht, weil die es konstruierenden Ingenieure dies verdrängt hätten, das ist eine zu einfache Antwort. Der Grund ist, dass die materialen Bedingungen der angewandten Mathematik als Informatik keine mathematische Null zulassen, die Mengen der Maschine enthalten keine Leeren Mengen, die Null ist ein positiver Zustand und nicht der Name der Leere (vgl. Heimann u. Hübener 2023). Denn jede Leere ist adressiert (sie liegt irgendwo im Speicher), jede Null ist typisiert (sie gehört einem Datentyp an), jedes ‚None‘ ist semantisch festgelegt (kein Ergebnis, aber doch ein Wert). Die Leere kann nicht oszillieren, nicht verschieben, nicht entziehen. Sie ist gefangen im Register der Positivität, das Nichts ist also eine Zahl unter anderen. Badious „umherwandernde Leere“ (vgl. Badiou 2005), also jene Leere, die sich in jeder Menge wieder als Unmarkierbares fortsetzt, würde in dieser Logik sofort stabilisiert: sie müsste als definierter Zeiger oder als *exception state* fixiert werden. Damit verliert sie ihren eigentlichen Charakter: nicht in der Ordnung, sondern an ihrem Rand zu stehen.

Was nicht bedeutet, dass die Maschinen nicht die symbolischen Regeln beherrschen würden, mit denen die Mathematik die Null zähmte. All diese Regeln funktionieren, aber ihr Fundament funktioniert nicht. Was bedeutet, dass die Psychose, oder die Cyberpsychose, nicht ein reduzierter Zustand ist, sondern der Zustand, in dem sich die Sprache des Computers immer bewegen muss, solange nicht die fundamentale Ontologie der Informatik verändert wird. Solange ‚Nichts‘ ein Zustand ist, gibt es kein Subjekt.

Der ‚Seepferdchen-Vorfall‘ ist so kein kurioser Ausrutscher künstlicher Intelligenz, sondern ein Symptom, in dem sich die Struktur von Transformer Modellen zeigt. Was sich zeigt, ist, dass die Sprache der Maschine den Mangel nicht denken kann, weil das Digitale das Nichts als Zustand, nicht als Leere begreift. Solange dies so bleibt, kreist jede Berechnung, jedes Verstehen um ein Loch, das sie nicht benennen kann. Allerdings weist Lacan auch darauf hin, dass wesentliche psychische Strukturen psychotisch strukturiert sind (Lacan 2016, S. 158-159) und die bekannten *Donut*-Strukturen Lacans (s. Lacan 2002, S. XV A, XVII 10-23) ermöglichen gerade das Kreisen um eine Abwesenheit, ohne die Abwesenheit zu repräsentieren. Vermutlich wäre der Weg, KI-Modelle tatsächlich weiterzuentwickeln, nicht zu machen, ohne solche Strukturen zu integrieren.

Verdrängte Seepferdchen.
Lustige KI-Psychose?

Literaturverzeichnis

Badiou, Alain (2005): *Das Sein und das Ereignis*. Berlin: diaphanes.

Brown, Tom B., Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared, Dhariwal, Prafulla et al. (2020): „Language Models are Few-Shot Learners“. In: *NeurIPS 2020*. DOI: 10.48550/arXiv.2005.14165.

Conneau, Alexis, Lample, Guillaume, Ranzato, Marc'Aurelio, Denoyer, Ludovic, Jégou, Hervé (2017): „Word Translation Without Parallel Data“. Published as a conference paper at ICLR 2018. DOI: 10.48550/arXiv.1710.04087.

García-Ferrero, Iker, Altuna, Begoña, Alvez, Javier, Gonzalez-Dios, Itziar, Rigau, German (2023): „This is not a Dataset: A Large Negation Benchmark to Challenge Large Language Models“. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, S. 8596-8615. DOI: 10.18653/v1/2023.emnlp-main.531.

Gubelmann, Reto; Handschuh, Siegfried (2022): „Context Matters. A Pragmatic Study of PLMs' Negation Understanding“. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, S. 4602-4621. DOI: 10.18653/v1/2022.acl-long.315.

Heimann, Marc (2024): „Der Spiegeloperator. Lacans Logik des Phantasma“. In: *Y – Zeitschrift für atopisches Denken* 4(3), 2.

Heimann, Marc, Hübener, Anne-Friederike (2023): „Material calculation and its unconscious: approaching computerization with Heidegger and Lacan“. In: *Psychoanalysis, Culture & Society*. DOI: 10.1057/s41282-023-00407-3.

Heimann, Marc, Hübener, Anne-Friederike (2024): „The extimate core of understanding: absolute metaphors, psychosis and large language models“. In: *AI & Society*. DOI: 10.1007/s00146-024-01971-7.

Lacan, Jacques (2002): *The Seminar, Book IX. Identification (1961-1962)*. Übers. von Cormac Gallagher. London: Karnac Books.

Lacan, Jacques (2016): *Die Psychose. Das Seminar, Buch III (1955-1956)*. 3. Auflage. Wien/Berlin: Turia + Kant.

McKenna, Nick, Li, Tianyi; Cheng, Liang, Hosseini, Mohammad Javad, Johnson, Mark, Steedman, Mark (2023): „Sources of Hallucination by Large Language Models on Inference Tasks“. DOI: 10.48550/arXiv.2305.14552.

Morante, Roser, Blanco, Eduardo (2021): „Recent advances in processing negation“. In: *Natural Language Engineering* 27 (2), S. 121-130.
DOI: 10.1017/S1351324920000534.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N. et al. (2017): „Attention Is All You Need“. In: *31st Conference on Neural Information Processing Systems*. DOI: 10.48550/arXiv.1706.03762.

Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Ichter, Brian, Xia, Fei et al. (2022): „Chain-of-Thought Prompting Elicits Reasoning in Large Language Models“. In: *36th Conference on Neural Information Processing Systems*. DOI: 10.48550/arXiv.2201.11903.

Autor:in: Marc Heimann ist Philosoph, spezialisiert auf philosophische Logik, beeinflusst von Heidegger und Lacan. Er promovierte 2020 an der Westfälischen Wilhelms-Universität Münster mit einer Arbeit über Heideggers Logik. Derzeit ist er wissenschaftlicher Mitarbeiter an der Hochschule Niederrhein, wo er sich auf die philosophischen Aspekte künstlicher Intelligenz konzentriert. Seine Forschung umfasst Projekte zu AI in den angewandten Sozialwissenschaften und die philosophische Untersuchung der digitalen Gesellschaft.

Kontaktinformation: marc.heimann@hs-niederrhein.de