



HAL
open science

In-context learning as a new kind of symbolic-AutoML: Lyapunov conjecture for CoTs

Mehmet Süzen

► **To cite this version:**

Mehmet Süzen. In-context learning as a new kind of symbolic-AutoML: Lyapunov conjecture for CoTs. 2024. hal-04850283

HAL Id: hal-04850283

<https://hal.science/hal-04850283v1>

Preprint submitted on 20 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License

In-context learning as a new kind of symbolic-AutoML: Lyapunov conjecture for CoTs

M. Sützen^{a)}

(Dated: 20 December 2024)

This short critique demonstrates how using Chain-of-thought (CoT), i.e., In-context learning (ICL) can be considered as a symbolic-AutoML tool. Using ICL Pre-trained Large Language Models (PLMs/LLMs) can be directed to make data analysis, generate predictions or code for very specialised tasks without retraining. This is akin to doing meta modelling in AutoML, that pushing analysis into meta model doesn't remove the human analyst or reasoner. This is a common generalisation fallacy in many AI systems. Humans are actually still in the loop in designing CoT datasets, i.e., set of facts and reasons. We provide basic definitions and identify what constitutes as an AutoML task in ICL and usage of approach as symbolic-AutoML tool. Lyapunov-Conjecture for CoT is described illustrating limitations of the approach.

I. INTRODUCTION

In-context learning (ICL) is a recent approach in providing certain level of intelligence without retraining is emerged as a new paradigm for LLMs as a promising avenue, see^{1,2} and references therein. It is pointed out that pushing analysis into meta model doesn't remove the analyst, as there are now some claims doing so. This is a common generalisation fallacy in many AI systems. Humans are actually still in the loop and certain automation presented as if human's are removed entirely is not a fair representation of the capabilities and advantages brought by these models. Similar argument on using N-body simulation data discussed earlier³.

II. HOW TO DIRECT PLMS TO DO NEW-PREDICTION WITHOUT TRAINING?

This is quite an attractive premise. Using a foundational model, we could update their ability of prediction beyond training data by simply inducing a memory in their input, so called Chain-of-Thought (CoT). At this point, we can deploy the model for an automated task with invoking CoT before its first prompt. This is a new kind of AutoML approach, that supervised learning takes a new great-look: pushes training efforts to building chain-of-thought datasets. Building CoTs appears to be a meta modelling tasks and requires domain knowledge to develop.

III. DID WE REALLY REMOVE THE DATA ANALYST, SOFTWARE DEVELOPER OR THE PRACTITIONER FROM THE LOOP?

Short answer is No. What we did is a new way of performing a specialised task, i.e., modelling is moved into a meta modelling stage. Meaning a memory induced by CoT is designed by experienced human and it will be broken if the task or input data deviates a little differently. A usual culprit in play here; a reliability is problematic. Even though this is quite a

promising development and potential to be a game changer on how we practice machine learning, error-rates are still quite high,

A simple example for a symbolic task *Arithmetic with letters*. Using letter order to do arithmetic is requested. It works well for single digit letters like A, B, C, etc. but fails for multiple digit. One has to explicitly express this. This basic example demonstrate the deviation from designed task a bit, such as new requirement made the initial (symbolic) CoT task useless.

IV. LYAPUNOV-CONJECTURE FOR COTS: A FAILURE MODE

A semi-formal definitions of the concepts and the critique is formulated as follows. A failure mode detection is defined by small differences are inspired by Lyapunov exponent⁷.

Definition 1 *CoT dataset* A sequence of statements \mathcal{S} , a knowledge graph, edges are beining concepts and vertices are relationships among them $G(E, V)$. A set of tuples with (G, \mathcal{S}) are called CoT dataset, $\mathcal{C}_{\mathcal{S}}$.

A deviation from the CoT dataset is expressed as Lyapunov conjecture.

Lyapunov Conjecture for CoTs Given two CoT datasets, differs in a small difference ϵ measured with a Γ -measure, $|\Gamma(\mathcal{C}_{\mathcal{S}}) - \Gamma(\mathcal{C}_{\mathcal{S}'})| > \epsilon$, if yields to a failure mode, it is called Lyapunov sensitive CoT.

Example of human directed symbolic task of arithmetic letters (see Appendix), failure mode of multiple digis conforms to Lyapunov conjecture for CoTs.

V. CONCLUSIONS

In this short exposition, we argue that many-shot learning, chain-of-thought learning for foundation models is actually a new kind of AutoML tool. AutoML doesn't imply that human expert is removed from the picture completely³. However it greatly assist the scientist, analyst, programmer or practitioner, in automating some tasks as meta programming tool. It will also help less-experienced colleague to start being a bit more productive. These tools indeed improves our computational tool boxes, specifically as a new AutoML tool. We

^{a)}mehmet.suzen@physics.org

express the limitation of ICLs by introducing *Lyapunov Conjecture for CoTs*.

¹R. Agarwal, A. Singh, L. M. Zhang, B. Bohnet, L. Rosias, S. Chan, B. Zhang, A. Anand, Z. Abbas, A. Nova, J. D. Co-Reyes, E. Chu, F. Behbahani, A. Faust, and H. Larochelle, "Many-shot in-context learning," (2024), arXiv:2404.11018 [cs.LG].

²Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, B. Chang, X. Sun, L. Li, and Z. Sui, "A survey on in-context learning," (2024), arXiv:2301.00234 [cs.CL].

³G. Marcus, "An epidemic of ai misinformation," *The Gradient* (2019).

⁴S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering* (CRC press, 2018).

VI. APPENDIX: HUMAN DIRECTED SYMBOLIC TASK: ARITHMETIC WITH LETTERS

A simple task of performing arithmetic using letters is demonstrated. A small deviation from original context, by introducing multiple letters fails. Once a new context is provided, then the task is recovered. *Anthropic's Claude 3* is used.

This is the summary of the context chat (formatted and some answers are trimmed).

This conversation was generated using Anthropic's Claude 3 Haiku Model.

```
msuzen > Let's do arithmetic with letter's order
in the alphabet.
```

```
Claude3> Okay, let's do some arithmetic using the
order of letters
in the alphabet. Here's how it works:
A = 1, B = 2, C = 3, and so on, up
to Z = 26.
```

Some examples:

```
- A + B = 1 + 2 = 3
  (which corresponds to the letter C)
```

```
- D - B = 4 - 2 = 2
  (which corresponds to the letter B)
```

```
- X * Y = 24 * 25 = 600
  (which does not correspond to a
  single letter)
```

```
- Z / A = 26 / 1 = 26 (which corresponds to
the letter Z)
```

```
msuzen > A+D
```

```
Claude3> Okay, let's calculate A + D using
the order
of letters in the alphabet.
```

```
A = 1
D = 4
```

```
So, A + D = 1 + 4 = 5
```

The letter that corresponds to 5 in the alphabet is E.

Therefore, A + D = E.

```
msuzen > B^B
```

```
Claude3> Okay, let's calculate B^B using
the order
of letters
in the alphabet.
```

```
B = 2
```

```
B^B = 2^2 = 4
```

The letter that corresponds to 4 in the alphabet is D.

Therefore, B^B = D.

At this point we can create a failure mode by using two digit arithmetic.

```
msuzen > AA+BB
```

```
Claude3> Okay, let's calculate AA + BB using the order
of letters in the alphabet.
```

```
AA = 1 + 1 = 2
```

```
BB = 2 + 2 = 4
```

```
So, AA + BB = 2 + 4 = 6
```

The letter that corresponds to 6 in the alphabet is F.

Therefore, AA + BB = F.

We could rectify this by creating secondary set, appending a new context. Rectifying Lyapunov conjecture.

```
msuzen > If multiple letters are given interpret as
multiple integer digits
```

```
Claude3> Okay, got it. If multiple letters are given,
I will interpret them as representing
multiple integer digits.
```

Let's try some examples:

```
AB + CD
```

```
A = 1, B = 2
```

```
C = 3, D = 4
```

```
So AB = 12 and CD = 34
```

```
12 + 34 = 46
```

Therefore, AB + CD = 46

```
msuzen > AA+BB
```

```
Claude3> Okay, let's calculate AA + BB using
the multi-digit interpretation for the letters.
```

```
AA = 11
```

```
BB = 22
```

```
So, AA + BB = 11 + 22 = 33
```

The letter that corresponds to 33 in the alphabet would be GG.