DOI: 10.6084/m9.figshare.32952530

An AI generated a novel hypothesis for you:

Curriculum-Induced Symbol Invention for Emergent Reasoning Primitives (CISI) – Hypothesis 2

Problem

LLMs passively absorb whatever tokens they see, but they are rarely encouraged to invent concise internal symbols that compress recurring patterns across tasks, leading to inefficiencies and poor generalisation to novel reasoning domains.

Motivation

A curriculum that gradually increases task complexity while imposing a discrete information bottleneck forces the model to coin and reuse its own internal ‘words,’ effectively creating a private domain-specific language (DSL) for reasoning.

Proposed Method

Curriculum-Induced Symbol Invention (CISI) extends standard pre-training with an auxiliary discrete codebook layer inserted between the transformer’s middle layers. During forward passes, hidden states are vector-quantised to the nearest codeword; the indices of these codewords act as latent symbols. A second decoder head tries to reconstruct the original hidden state from just the indices, adding a strong compression pressure.

We design a staged training curriculum: Stage 1 uses tiny synthetic tasks (single-digit addition, simple relational statements) where the minimal set of 10–20 codewords suffices. Stage 2 introduces compound tasks requiring composition; the previous codewords become building blocks. The curriculum continues up to real GSM8K and CommonsenseQA. At each stage the codebook may expand slightly, but sparsity regularisers keep growth slow, encouraging reuse.

DOI: 10.6084/m9.figshare.32952530

We periodically probe the learned code indices with clustering techniques and linear probes, translating them into human-interpretable tokens like “ADD,” “COMPARE,” “OLDER_THAN.” Optionally we align them with Unicode glyphs to create a visible notation. The main LLM can then be prompted to emit sequences of these invented symbols alongside natural language, offering transparent intermediate steps.

Because these symbols emerge organically from data efficiency pressures rather than hand engineering, they are expected to be well-aligned with the statistical structure of language and transfer well across domains.

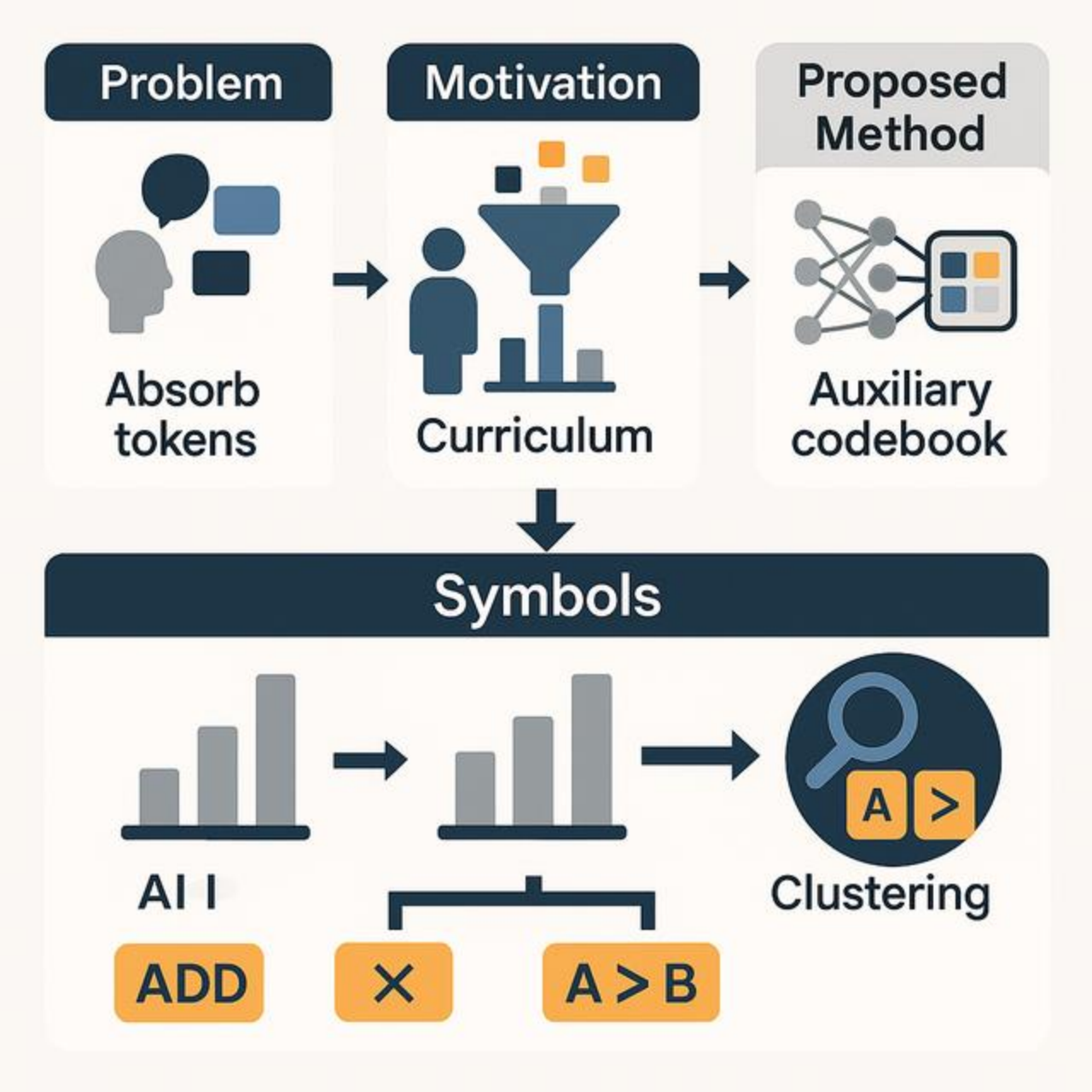
Experiment Plan

1. Track codebook utilisation, entropy, and growth across the curriculum.
2. Evaluate zero-shot transfer to new reasoning datasets versus baseline models of equal parameter count.
3. Perform ablation by disabling the compression layer, measuring drop in generalisation.
4. Human study: present the emergent symbols and ask annotators to map them to English descriptions; measure consistency agreement.

Non-Expert Description

Think of teaching a child math by first letting them count pebbles, then sticks, then finally inventing the + sign to save time. Our method guides an AI through a similar journey: we start with easy puzzles and slowly make them tougher, all while squeezing the AI so it can’t store every detail. To cope, it makes up its own shorthand marks—its personal “+,” “×,” or “is older than.” By the end, the AI has a pocketful of handy mini-words that let it solve brand-new puzzles faster, much like a seasoned mathematician scribbles quick symbols instead of whole sentences.

DOI: 10.6084/m9.figshare.32952530



How we generate hypotheses for you:

1. Our system digests 100 million peer-reviewed papers to build the knowledge base.
2. A large language model then uses this knowledge base to generate 92,000 draft ideas, using about 30 million tokens of generation.
3. The drafts face 23,000 rapid AI match-ups to choose the top 50 ideas—a 0.05% acceptance rate.

https://www.academia.edu/ai_hypothesis/QkDOgN_GGLoOP