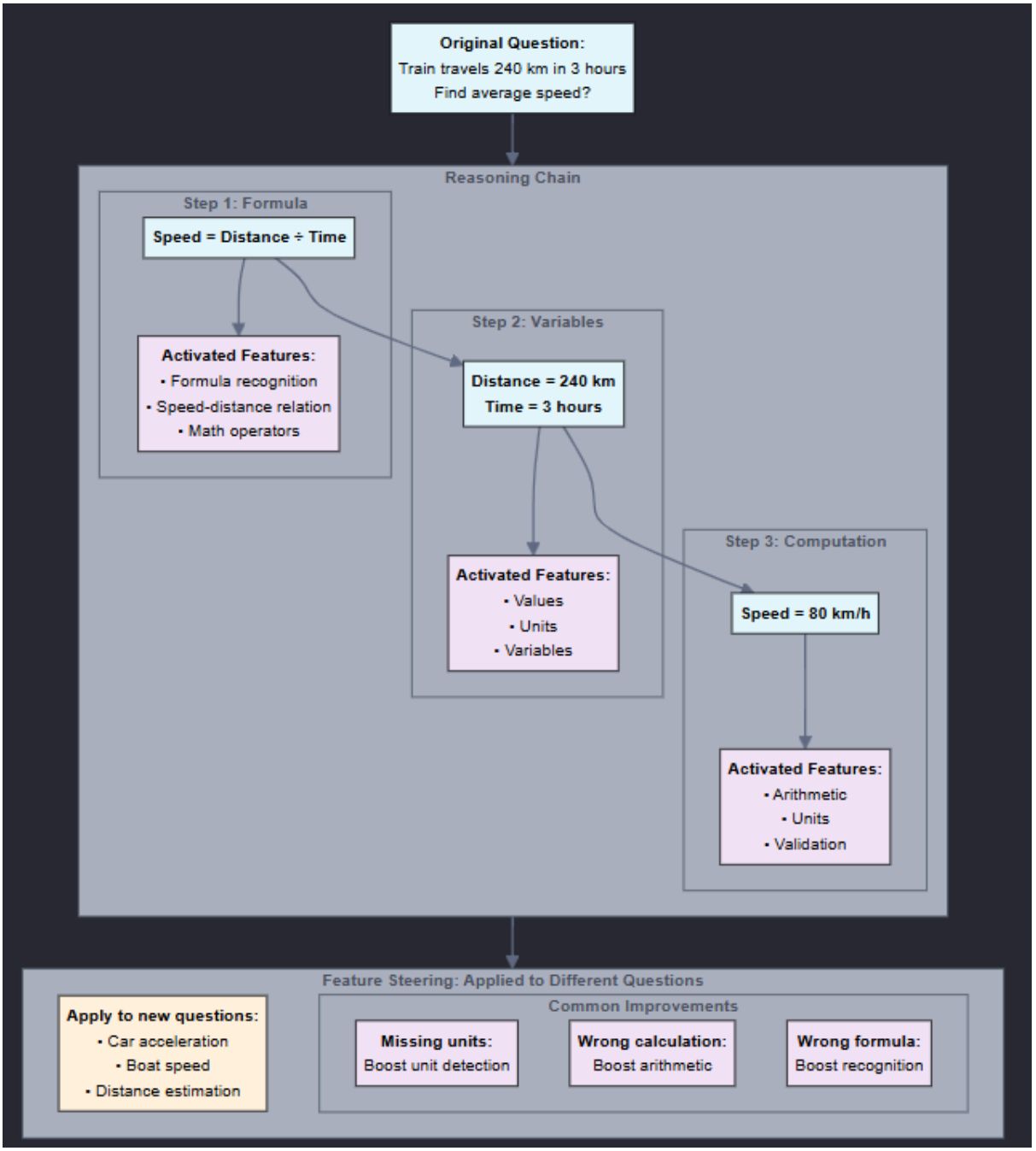
Natural Language Reinforcement Learning via SAE Features

1. Decompose answer outputs into COT reasoning chains
2. Identify features associated with correct reasoning chains,
3. Use these features to steer incorrect reasoning chains into correct reasoning chains
4. RL/finetune on the correcting reasoning outputs



Setup

First, we take:

- TARGET model - Model which we plan to do RL/finetuning on, for which we have a set of reconstructed SAE features. For this setup. We will assume the use of Goodfire AI tools, since they do a lot of setup work and have also implemented some of the steps mentioned here.
- EXAMINER model - A second model which is unchanged and will act as an agent to conduct follow-up question, interpretability experiments and probing. This model probably doesn't have to be bigger/more advanced than the TARGET model.

Steps

Query selection:

1. **Choosing KNOWN queries to test** Given a set of queries on a given topic, we set aside some queries as KNOWN queries for training/finetuning, where we give the model access to the answer/solution set. We set aside UNKNOWN queries for testing. Since we want reasoning gains to generalise, we can actually randomise this process.
    a. Note that this is how O1 finetunes are done - there has to be a known and unknown set: Reinforcement Fine-Tuning—12 Days of OpenAI: Day 2 - YouTube
2. **Answer Verification** For KNOWN queries, we test the TARGET model on a dataset with known answers. We would probably expect to find CORRECT and INCORRECT answer output.

COT Decomposition: Full answer output -> Sentence-level COT -> Token-Level

3. **Decompose COT steps of answer:** First, given an answer to any KNOWN query we decompose each step of the reasoning process for verification. This is established to be optimal for longer complex COT on longer, harder problems. Examples of step-by-step reasoning approaches
    a. [2305.20050] Let's Verify Step by Step
    b. [2306.04031] Certified Deductive Reasoning with Language Models
4. **Identify critical tokens within COT steps within answers:** We identify which specific reasoning tokens within the reasoning step contribute the most to achieving CORRECT KNOWN and INCORRECT UNKNOWN reasoning chains. This is done by comparing with the ground truth dataset correct explanations.
    a. [2411.19943] Critical Tokens Matter: Token-Level Contrastive Estimation Enhances LLM's Reasoning Capability
    b. *[Optional] EXAMINER AGENT: It probably also helps to get the EXAMINER model to elicit COT to ask clarifying questions that expand explanations of tokens and how they contribute to answers. Think of it as COT^2.*

[2412.15838] Align Anything: Training All-Modality Models to Follow Instructions with Language Feedback

c.

Feature analysis: Attribution from answer outputs to features

5. **Search for the feature:** We identify which features activate on KNOWN answer tokens, both CORRECT KNOWN answers and INCORRECT KNOWN answers.
    a. Goodfire documentation mentions Intervention and Attribution as a possible mechanism for this. See **Examples of feature explanation and isolation:**
        i. Understanding and Steering Llama 3 with Sparse Autoencoders - Goodfire Papers
        ii. Open Source Automated Interpretability for Sparse Autoencoder Features | EleutherAI Blog
        iii. **IMPLEMENTATION:** Sorting by Features - Goodfire SDK
        iv. **IMPLEMENTATION:** Conditionals - Goodfire SDK - for COT
    b. *[Optional] EXAMINER AGENT: Using Expanded COT^2 could also allow us to dynamically expand, interrogate and contrast the reasoning process. For example, instead of a 10-sentence answer with 100 features, Expanded COT^2 can give you 100-sentence answers with 1000 features.*
    c. IMPLEMENTATION: Features - Goodfire SDK
    d. **IMPLEMENTATION:** Conditionals - Goodfire SDK

Feature Steering: Iterative testing using SAE feature steering

6. On UNKNOWN queries, for INCORRECT outputs, again identify critical tokens and steer/turn on features associated with CORRECT KNOWN answer outputs.
    a. Understanding and Steering Llama 3 with Sparse Autoencoders - Goodfire Papers
    b. **IMPLEMENTATION:** Variants - Goodfire SDK
7. **RL/Finetuning** until the model gets the answer correct
    a. [NOTE: Unclear what the best RL/finetuning method is for this step]
    b. It is likely also possible to do RL on the feature/latent activations themselves instead of the outputs associated with feature/latent activations, but I haven't figured out how to do this well.
8. **Fuzzing features on rephrased queries:** To ensure reasoning robustness/generalisation, you may repeat this with multiple associated features, multiple rephrasings, multiple COT outputs for the same question and multiple associated questions.
    a. **Averaging** To minimise sensitivity to rephrasing, you could simply generate multiple answers with the steered CORRECT features and average out the vectors for all CORRECT/acceptable answers.
    b. **IMPLEMENTATION:** Auto Steer - Goodfire SDK

Over time, the TARGET model should get better at answering queries.

Why use SAE features with COT instead of pure text with COT?

Natural language flexibility Steering features/latents lets you adjust/fuzz/iterate answers much more flexibly than changing tokens, especially in decomposed COT. Tokens would probably work on deterministic problems like maths, but less so on complex COT reasoning or less deterministic natural language domains.

Replicabiity/robustness to rephrasing Features are more robust to rephrasing, variable-length ablation and complex natural-language queries like we'd want in COT.

> It is also possible to do **Averaging** of rephrased correct answers withthe same features.

Capturing complex concepts Features can also capture more complex/abstract non-token or mutli-token concepts.

> Context length is a known problem for existing SAE feature reconstructions. Might be a problem especially for really long COT, for which we'd either have to solve or find smart COT decomposition hacks around [Examining Language Model Performance with Reconstructed Activations using Sparse Autoencoders — LessWrong]
>
> Possible solution: Entropy-based batching or leveraging RoPE or attention from LLMs, or the LLMs themselves.

Faithfulness - The inherent problem with COT finetuning is you're never sure if the model is actually reasoning. If it's simply copying the tokens, then this would not improve, and possibly even harm reasoning. Since we're using SAEs and can vary the phrasing many different ways, token bias is less of a concern and issue and we have somewhat greater confidence of capturing concepts beyond the token level.

Re: Capabilities Concerns

As you can see, most of the proposed implementation steps have already been published and applied elsewhere to improve LLM reasoning. I expect the O1 team has tried/implemented most of these and more. The main novel contribution is the use of SAE features to improve reasoning. If SAEs prove useful in improving reasoning models, then we will see frontier reasoning paradigms tied in with SAEs. If it's not that useful, then we aren't really introducing new capabilities concepts anyway.

Random notes:

Fuzzing: Scaling Automatic Neuron Description | Transluce AI

Attribution: Understanding and Steering Llama 3 with Sparse Autoencoders - Goodfire Papers

Conditionality/causality - Do we need conditional COF? We are limited by cross layer and causative fuzzing

Data - Need to manage data tradeoffs - find better data management techniques

Does it need to be features - i actually think it does because it prevents overindexing on token-level correlations.

Deconstruct token?

Agent feature testing: Let LLM Agents Perform LLM Surgery

COT specific features: reprogramming_ai_models/report.pdf at main · ThomasWalker1/reprogramming_ai_models

Answer testing

Adaptive threshold
Blt like clustering

Problem: Reinforcement Learning approaches to improve natural language reasoning

1. Decompose outputs into COT statements
2. Identify features involved in statements/tokens and boost features that contribute to correct reasoning chains/answer tokens

Invariance + mass dominance + truncating tails discriminatively

- If there's no clear, it doesn't matter
- If there is, it does

https://transluce.org/neuron-descriptions

COT decomposition:

O1
Critical token

SAEs

Theoretically, as long as we get a reasonable scalar value of a feature's contribution in an output, . Pretty much all existing SAEs do this.

Evidence of RL with logprobs

SimPO
SSRM
SPA

Self-evaluation

(3) Séb Krier on X: "LLMs can self-improve through generating possible answers, verifying them, and subsequently re-weighting the importance of its generated answers. This paper introduces GV-Gap, which measures the "precision" of a model's self-verification. The authors find that the larger the https://t.co/DIMTbzegjA" / X
Rohan Paul on X: "Synthetic data and iterative self-improvement is all you need. No humans needed in the evaluation loop. This paper introduces a self-improving evaluator that learns to assess LLM outputs without human feedback, using synthetic data and iterative self-training to match top https://t.co/g9oDvRKF35" / X