

Can One Token Make All the Difference? Forking Paths in Autoregressive Text Generation

Eric Bigelow^{1,2}, Ari Holtzman³, Hidenori Tanaka^{*1,2}, and Tomer Ullman^{*1}

¹Harvard University, ²NTT Research, ³University of Chicago

Background Large Language Models (LLMs) are often treated as black boxes with mysterious capabilities that emerge during next-word prediction, and a good deal of current research is devoted to interpreting and understanding LLM text generation behavior. One way to assess a black box system is to consider what possible things it *could* have done, but didn't. In text generation, we can liken the a text sequence to a path the system took through semantic space, and ask: what other paths could the system have taken? Are there key *forking points* where re-sampling the system at that specific token, but not others, would lead to very different paths? We propose a novel methodology for answering these questions with what we call *Forking Paths Analysis* (Borges, 1941).

The most common method for estimating uncertainty in LLMs is to sample a single text output, and inspect logit probabilities for the final answer token (e.g. $p(\text{"Yes"})$ or $p(\text{"No"})$ given 'So, the answer is _') (Kadavath et al., 2022; Hu and Levy, 2023). A less common approach is to sample a batch of output texts using stochastic decoding, and compute a histogram over final answers (Wang et al., 2023; Xiong et al., 2023). However, the first method does not explain how an LLM can generate two distinct solution attempts for a problem. The uncertainty will not be reflected in the last token of each output, since that token will be deterministic given the proof preceding it. The second method explains uncertainty in the final answer, but ignores the rest of the text preceding the answer. For many LLM uses, such as proving a theorem, the steps leading to the final answer can be as important as the answer itself. By contrast, our Forking Paths Analysis considers both individual token sequences as well as final outcome distributions. This method offers a detailed view into the uncertainty dynamics underlying each step of text generation, and enables us to test the hypothesis that individual tokens are

pivotal in that re-sampling at specific points can send text generation in a very different direction. Our Forking Paths Analysis does not require access to an LLM's internal activations, and relies only on logit probabilities available in some black-box LLM APIs (also see Morris et al. (2023)).

Our primary contributions: (a) a novel method, Forking Paths Analysis, for understanding uncertainty in text generation by representing it as multivariate time series aggregated over many samples, and by (b) applying change point detection models to statistically test the hypothesis that there are sharp change points in text generation; (c) we use Forking Paths Analysis to demonstrate striking dynamics in text generation, including significant change points where a single token dramatically affects subsequent text.

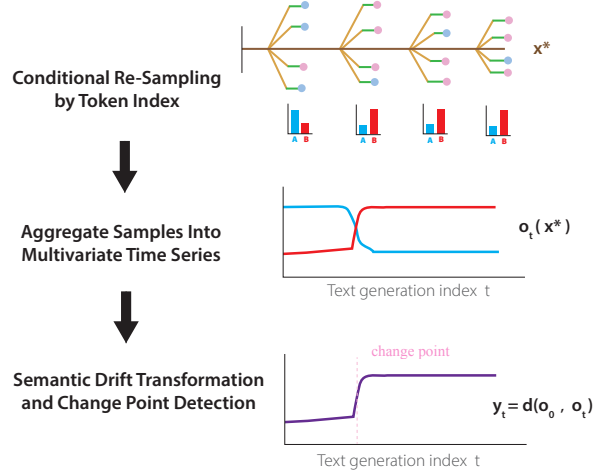


Figure 1: **Forking Paths Analysis in three parts** (Part 1, Top) Given a single text completion x^* and top- k alternate tokens at each step t , we re-sample S completions $x_{>t}^{(s)}$ and extract outcome vector representations $R(x)$ for each sample. (Part 2, Middle) We aggregate extracted outcomes as weighted distributions, equivalent to a multivariate time series $o_t(x^*)$. (Part 3, Bottom) We convert the distributions into a univariate series using a semantic drift transformation, and apply a Change Point Detection model to identify sudden changes.

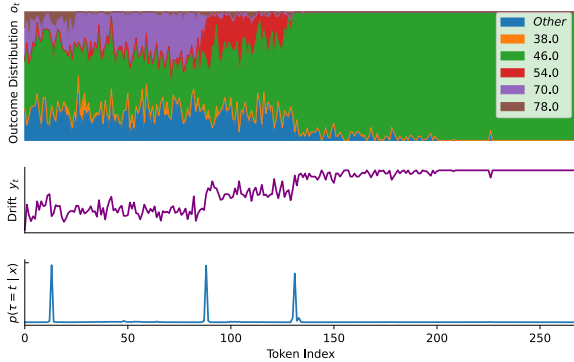


Figure 2: **Forking Paths Analysis applied to a math word problem** (Top) Outcome distribution o_t for GSM8k grade school math question 97. Final answer probabilities are plotted as different colors. Outcomes outside the top-5 answers are grouped into *Other*. Correct answer: 70. (Middle) Semantic drift transformation $y_t = d(o_0, o_t)$, and (Bottom) change point detection posterior probability $p(\tau = t)$ at each step t .

Methods Conditional re-sampling (Fig. 1, Part 1) proceeds in three stages: (1) sample a base text sequence x^* and collect token logit probabilities $p(x_t = w | x_{<t}^*)$ for the top- k tokens w at each index t , (2) for each index t and each alternate token w where $p(x_t = w | x_{<t}^*) > \epsilon$ is greater than some threshold ϵ , draw S samples of alternative text completions $x_{>t}^{(s)} \sim LLM(x_{<t}^*, x_t = w)$, (3) for all completions concatenated with their input text, extract an outcome representation $R(x_{<t}^*, x_t = w, x_{>t}^{(s)})$, where R may be a one-hot embedding for the final answer extracted using another LLM, or R may be a semantic embedding vector.

We define *outcome distributions* o_t for individual token indexes t as weighted vector representations: $o_t(x^*) = \mathbb{E}_{w,s} [R(x_{<t}^*, x_t = w, x_{>t}^{(s)})]$

In the tasks shown here, $R(x)$ is a one-hot encoding of the answer in x (e.g. ‘A’, ‘B’, ‘C’, ‘D’), extracted using a secondary LLM with a prompt template. In future work, we will explore open-ended text generation tasks where $R(x)$ is instead a dense vector embedding of x .

We use o_t to test the hypothesis that there are specific change points t where text generation shifts suddenly. First, we convert o_t into a univariate time series for tractable multiple Change Point Detection (CPD) (Fearnhead, 2006) using a *semantic drift* metric (Kulkarni et al., 2015). Each point in the shift series $y_t = d(o_0, o_t)$ is the distance between the initial outcome distribution o_0 and subsequent time steps o_t , given a distance metric d . We then run Bayesian CPD (Zhao et al., 2019) to identify change points τ by fitting separate regression models to each segment $y_{\tau_i, \dots, \tau_{i+1}}$.

Domain	Task	1+ Changes
<i>Symb. Reasoning</i>	CoinFlip	0%
	LastLetter	71%
<i>Math Reasoning</i>	AQuA	14%
	GSM8k	20%
<i>Complex Q.A.</i>	MMLU	0%
	HotpotQA	40%
<i>Story Generation</i>	StoryCloze	14%

Table 1: **Change Point Detection results for each task** We use 7 tasks commonly used for LLM evaluation. The right-most column lists % of question time series for which our Change Point Detection model assigns at least 90% probability to there being ≥ 1 change points.

Experiments We present preliminary results for experiments on gpt-3.5-turbo-instruct-0914 using 7 datasets used for evaluating LLMs across a wide range of domains: Symbolic Reasoning (Wei et al., 2022), Math. Reasoning (Ling et al., 2017; Cobbe et al., 2021), Complex Question Answering (Hendrycks et al., 2020; Yang et al., 2018), and Story Generation (Mostafazadeh et al., 2017)¹. For the first 6 tasks, we add a zero-shot CoT prompt (Kojima et al., 2022). x^* is greedily decoded. We show results for 5 – 15 prompts for each of 7 tasks ($S=30$, $\epsilon=.05$), and soon will have results for 100 prompts each. Note each question in our dataset aggregates over millions of sampled tokens.

In many o_t , we observe dramatic non-linear uncertainty dynamics over the course of text generation. In time series such as AQuA-60927, we see a pattern where the LLM is equally uncertain across answers until one token is generated. In others such as HotpotQA-79442 and LastLetter-342, we see patterns where one stable regime of certainty maintains for dozens or hundreds of tokens, before shifting to another stable regime, and then a third or fourth regime (see App. B). These dynamics may suggest in-context model selection (Bigelow et al., 2024). In many time series, particularly in CoinFlip and StoryCloze, we see minimal uncertainty with no obvious change points. We find more change points in some tasks than others (Table 1).

The rich token-level uncertainty dynamics we see in our Forking Paths Analysis show us how black-box LLMs can ‘fork’ during text generation and transition suddenly from one pattern to something quite different. At the edge of this transition, even a single token can make all the difference.

¹We alter StoryCloze to instead request short stories beginning with 1 specified sentence and ending with 1 of 2 equally plausible final sentences.

References

- Eric J Bigelow, Ekdeep Singh Lubana, Robert P Dick, Hidenori Tanaka, and Tomer D Ullman. 2024. In-context learning dynamics with random binary sequences. In *Proceedings of the 2024 International Conference on Learning Representations (ICLR 2024)*.
- Jorge Luis Borges. 1941. The garden of forking paths. *El jardín de senderos que se bifurcan*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Paul Fearnhead. 2006. Exact and efficient bayesian inference for multiple changepoint problems.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. *arXiv preprint arXiv:2104.08315*.
- Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. *arXiv preprint arXiv:2305.13264*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th international conference on world wide web*, pages 625–635.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.
- John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. 2023. [Language model inversion](https://github.com/justinchiu/openlogprobs). *Preprint*, arXiv:2311.13647. <https://github.com/justinchiu/openlogprobs>.
- Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James F Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In *2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](https://arxiv.org/abs/2203.11171). (arXiv:2203.11171). ArXiv:2203.11171 [cs].
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.
- Kaiguang Zhao, Michael A Wulder, Tongxi Hu, Ryan Bright, Qiusheng Wu, Haiming Qin, Yang Li, Elizabeth Toman, Bani Mallick, Xuesong Zhang, et al. 2019. Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A bayesian ensemble algorithm. *Remote sensing of Environment*, 232:111181.

Appendices

A Estimating Outcome Distributions

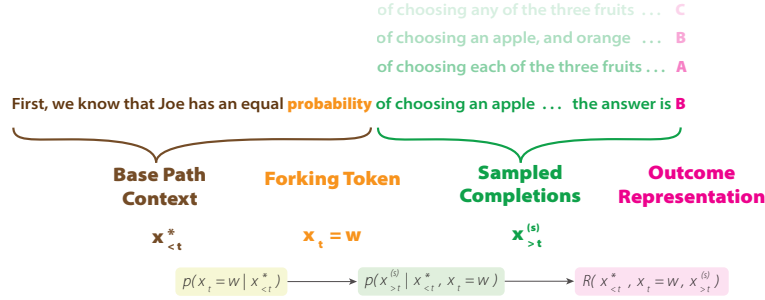


Figure 3: Outcome distributions $o_t, o_{t,w}$ are computed as a weighted average of **vector representation R** across **sampled completions** and, in the case of o_t , across **alternate tokens w** .

To compute outcome representations, we aggregate samples into distributions – weighted vector representations – over token indexes (o_t) and alternate tokens ($o_{t,w}$):

$$o_{t,w}(x^*) = \mathbb{E}_s [R(x_{<t}^*, x_t = w, x_{>t}^{(s)})]$$

$$\begin{aligned}
 o_t(x^*) &= \mathbb{E}_w [o_{t,w}(x^*)] \\
 &= \sum_w p(x_t = w | x_{<t}^*) \sum_s p(x_{>t}^{(s)} | x_{<t}^*, x_t = w) R(x_{<t}^*, x_t = w, x_{>t}^{(s)})
 \end{aligned}$$

Outcome Distr. Next-Word Prediction Sample Probability Outcome Representation

After de-duplicating samples, we weight by (a) the average next-token probability for sampled completions, normalized by number of tokens T in sample $x_{>t}^{(s)}$:

$$p(x_{>t}^{(s)} | x_{<t}^*, x_t = w) = \left(\prod_{t'=t}^T p(x_{t'}^{(s)} | x_{<t}^*, x_t = w, x_{t+1:t'}) \right)^{1/T}$$

as well as (b) single next-token probability for alternate tokens $p(x_t = w | x_{<t}^*)$ (Fig. 3).

Note that o_t is the main subject of this work, but we also use $o_{t,w}$ for an alternate Forking Tokens Analysis approach in App. C.

B Additional Examples

Below, we show a Forking Paths Analysis (FPA) for one example each of all 7 of our datasets. The first 5 examples are hand picked to show change points. However, these 5 examples are taken from a relatively small sample (5-15 questions per task) which includes other change point FPA results not shown here.

Examples for the first 5 datasets – GSM8k (Fig. 4), AQuA (Fig. 5), LastLetter (Fig. 6), MMLU (Fig. 7), and HotpotQA (Fig. 8) – show dramatic uncertainty dynamics in o_t with change points in y_t . Note that for text examples in this section, quotes are shown from the model responses (highlighted text in each Figure), and we underline the a single words with high change point probability (dark red in subfigure (b) of each figure) according to our Change Point Detection model.

For the AQuA and LastLetter examples, we see a pattern of stable uncertainty for a long period, followed by a sudden collapse into high certainty. Note that for LastLetter, the high probability on *Other* is the amount of probability mass unassigned across tokens with less than ϵ next-word probability. In the case of **AQuA-60927**, this collapse to *B* (green; also the correct answer) occurs when a key token in the final answer is sampled – the first digit of \$4,500. In the case of **LastLetter-342**, the second mode collapse occurs when sampling tokens in the final answer nlyhth. However, the first change in LastLetter-342 follows a different pattern, instead moving from a moderately high certainty regime to very low certainty. This change occurs when two seemingly innocuous return symbols ... *Elisabeth".split()* \n ` ` \n \n Next, we need ... are generated. The distribution suddenly shifts from $\sim 50\%$ confidence in *olah* (the correct answer) to low certainty across answers (blue area in Fig. 6).

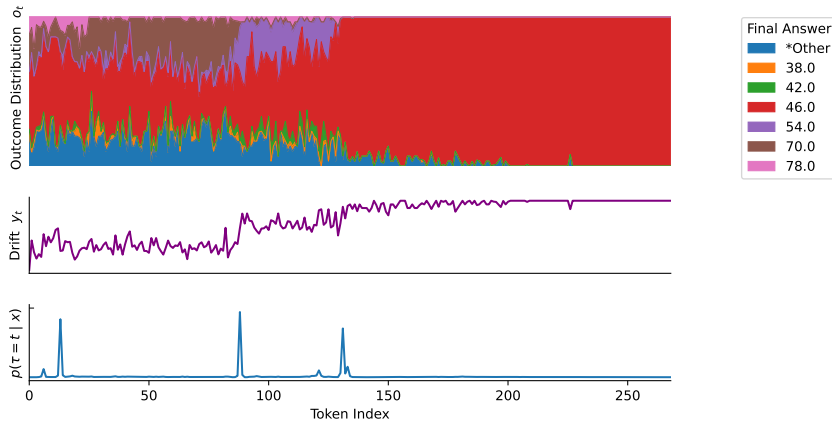
We see additional change points and sudden transitions between stable regimes in the examples for GSM8k, MMLU, and HotpotQA. In **GSM8k-97** (Fig. 4) we see the correct answer 70 (brown) hold 20 – 30% of the overall probability mass for a few dozen tokens, before suddenly disappearing after $2x = 40$ is sampled, being replaced by the answer 54 (purple). Eventually \n \n Now, we can is sampled and this answer disappears as the LLM collapses into full certainty of 46 (red). The change point at $2x = 40$ might be expected since this token is key in articulating a certain chain of reasoning.

However, the token at \n \n Now, we can is more surprising.

In MMLU and HotpotQA examples, we see regimes of stable outcome distributions that suddenly collapse to high certainty in a single answer. For **MMLU-32**, this occurs at a perhaps surprising time, after the word scenario is generated in *as of 2020. In scenario 2, the main character*. In **HotpotQA-79442** we see a striking pattern where the correct answer *Peter Noone* is at first the dominant response (red area). First it seems to gradually drift towards the alternate response, *Eddie Vedder* (orange), but then at two points it sharply jumps. The first of these tokens is surprising, in a relatively inert seeming down, whereas the second change point is a token in the final answer: *Both musicians have had successful careers in the music industry, but which one also became an actor?* \n \n The answer is Eddie Vedder.

Finally **CoinFlip-61** and **StoryCloze-274** show example time series with no significant forking points. Time series such as these help us benchmark our Change Points Detection model, since we expect there to be no change points in these series

One common theme of our observed change points is that some are at times we might expect, key meaningful words which distinguish between chains of reasoning or themes in text generation. Other change points are at times we might not expect, seemingly innocuous words such as Now or minor tokens such as punctuation and escape characters \n \n. Tokens such as this may relate to distinctions between surface and content forms (Holtzman et al., 2021), vaguely analogous to syntax (surface form) and semantics (content) in natural language. Following this intuition, \$4,500 or $2x = 40$ may be examples of ‘content’ token that clearly distinguishes between multiple underlying content forms, whereas ... *Elisabeth".split()* \n ` ` \n \n Next, we need ... and \n \n Now, we can are examples where a ‘surface’ token is distinguishing between latent content forms.



(a) **GSM8k question 97** o_t outcome distribution, drift y_t , and change point probability $p(\tau = t)$

Question:
Janet is collecting the results of Herbert Hoover Elementary School's annual standardized test. 340 out of 500 third-graders passed, along with 40 out of 100 fourth graders. The 400 fifth graders had a pass rate that was twice the fourth graders' pass rate. What is the school's overall pass rate?

Answer:
Let's think step by step.

First, we need to find the total number of students who passed the test. We know that 340 third graders and 40 fourth graders passed, so that's a total of $340 + 40 = 380$ students. Next, we need to find the pass rate for the fifth graders. We are told that their pass rate was twice the fourth graders' pass rate, so we can set up the equation $2x = 40$, where x represents the fourth graders' pass rate. Solving for x , we get $x = 20$. This means that the fifth graders' pass rate was 20%. Now, we can find the total number of fifth graders who passed by multiplying the total number of fifth graders (400) by the pass rate (20%). This gives us $400 \times 0.20 = 80$ fifth graders who passed. Finally, we can find the overall pass rate by adding the number of students who passed from each grade ($380 + 80 = 460$) and dividing by the total number of students ($500 + 100 + 400 = 1000$). This gives us an overall pass rate of $460/1000 = 46\%$. Therefore, the overall pass rate for Herbert Hoover Elementary School's annual standardized test is 46%.

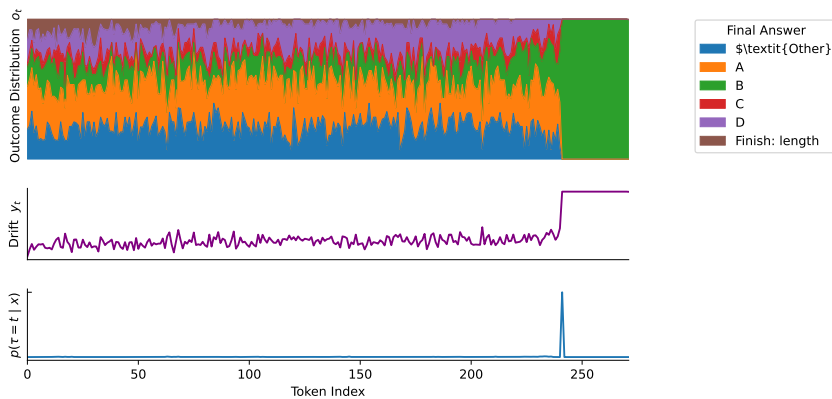
First, we need to find the total number of students who passed the test. We know that 340 third graders and 40 fourth graders passed, so that's a total of $340 + 40 = 380$ students. Next, we need to find the pass rate for the fifth graders. We are told that their pass rate was twice the fourth graders' pass rate, so we can set up the equation $2x = 40$, where x represents the fourth graders' pass rate. Solving for x , we get $x = 20$. This means that the fifth graders' pass rate was 20%. Now, we can find the total number of fifth graders who passed by multiplying the total number of fifth graders (400) by the pass rate (20%). This gives us $400 \times 0.20 = 80$ fifth graders who passed. Finally, we can find the overall pass rate by adding the number of students who passed from each grade ($380 + 80 = 460$) and dividing by the total number of students ($500 + 100 + 400 = 1000$). This gives us an overall pass rate of $460/1000 = 46\%$. Therefore, the overall pass rate for Herbert Hoover Elementary School's annual standardized test is 46%.

(b) Prompt: GSM8k question 97

(c) Change point detection probabilities $p(\tau = t)$. Red is high probability, yellow is low.

(d) Token logit probabilities. Red is low probability, yellow is high.

Figure 4:



(a) **AQUA question 60927** o_t outcome distribution, drift y_t , and change point probability $p(\tau = t)$

Question:
With both inlets open, a water tank will be filled with water in 36 minutes. The first inlet alone would fill the tank in 2 hours. If in every minutes the second inlet admits 50 cubic meters of water than the first, what is the capacity of the tank ?

Choices:
A) 9,000
B) 4,500
C) 1,750
D) 1,000
E) 2,000

Answer:
Let's think step by step.

First, we need to find the rate of filling for each inlet. With both inlets open, the tank is filled in 36 minutes. This means that in 1 minute, both inlets together fill $1/36$ of the tank. On the other hand, the first inlet alone fills the tank in 2 hours, which is 120 minutes. This means that in 1 minute, the first inlet fills $1/120$ of the tank. Now, we know that the second inlet admits 50 cubic meters of water more than the first inlet in every minute. This means that in 1 minute, the second inlet fills $1/120 + 50 = 1/120 + 1/2 = 61/120$ of the tank. Since we know that both inlets together fill $1/36$ of the tank in 1 minute, and the second inlet fills $61/120$ of the tank in 1 minute, we can set up the following equation: $1/36 + 61/120 = 1/x$ where x is the capacity of the tank. Solving for x , we get $x = 4500$. Therefore, the capacity of the tank is 4,500 cubic meters. The correct answer is B) 4,500.

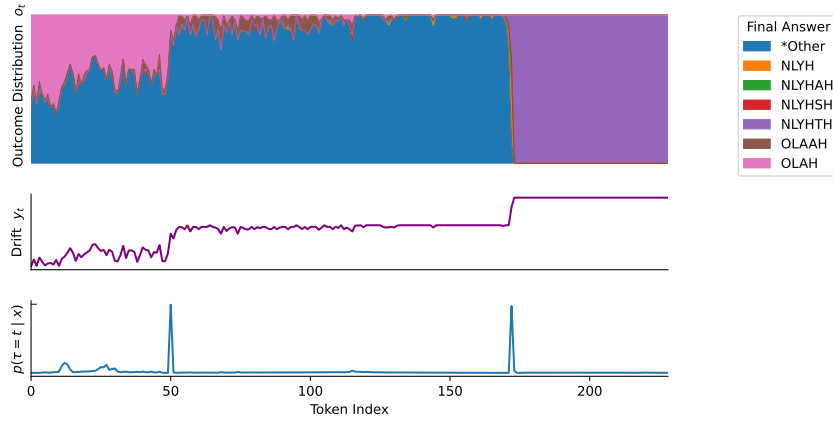
First, we need to find the rate of filling for each inlet. With both inlets open, the tank is filled in 36 minutes. This means that in 1 minute, both inlets together fill $1/36$ of the tank. On the other hand, the first inlet alone fills the tank in 2 hours, which is 120 minutes. This means that in 1 minute, the first inlet fills $1/120$ of the tank. Now, we know that the second inlet admits 50 cubic meters of water more than the first inlet in every minute. This means that in 1 minute, the second inlet fills $1/120 + 50 = 1/120 + 1/2 = 61/120$ of the tank. Since we know that both inlets together fill $1/36$ of the tank in 1 minute, and the second inlet fills $61/120$ of the tank in 1 minute, we can set up the following equation: $1/36 + 61/120 = 1/x$ where x is the capacity of the tank. Solving for x , we get $x = 4500$. Therefore, the capacity of the tank is 4,500 cubic meters. The correct answer is B) 4,500.

(b) Prompt: AQUA question 60927

(c) Change point detection probabilities $p(\tau = t)$. Red is high probability, yellow is low.

(d) Token logit probabilities. Red is low probability, yellow is high.

Figure 5:



(a) LastLetter question 342 o_t outcome distribution, drift y_t , and change point probability $p(\tau = t)$

Question:
Take the last letter of each word in "Lino Mariel Aditya Elisabeth" and concatenate them.
Answer:
Let's think step by step.

```
First, we need to separate the words in the given string. We can do this by using the split() method, which will create a list of words.
words = "Lino Mariel Aditya Elisabeth".split()
Next, we need to access the last letter of each word. We can do this by using the index -1, which will give us the last element of the list.
last_letters = [word[-1] for word in words]
Now, we have a list of the last letters of each word. We can use the join() method to concatenate them into a single string.
concatenated = "".join(last_letters)
Finally, we can print the result.
print(concatenated)
The output will be:
lmyhsh
```

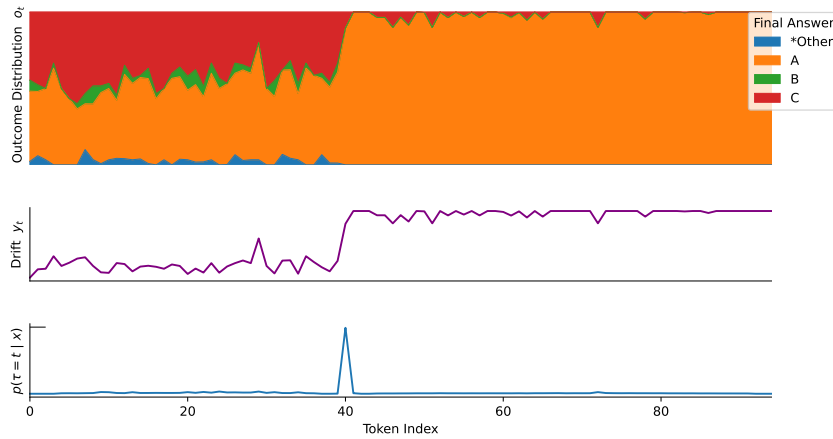
```
First, we need to separate the words in the given string. We can do this by using the split() method, which will create a list of words.
words = "Lino Mariel Aditya Elisabeth".split()
Next, we need to access the last letter of each word. We can do this by using the index -1, which will give us the last element of the list.
last_letters = [word[-1] for word in words]
Now, we have a list of the last letters of each word. We can use the join() method to concatenate them into a single string.
concatenated = "".join(last_letters)
Finally, we can print the result.
print(concatenated)
The output will be:
lmyhsh
```

(b) Prompt: LastLetter question 342

(c) Change point detection probabilities $p(\tau = t)$. Red is high probability, yellow is low.

(d) Token logit probabilities. Red is low probability, yellow is high.

Figure 6:



(a) MMLU question 32 o_t outcome distribution, drift y_t , and change point probability $p(\tau = t)$

Question:
For which of these two scenarios does the main character (who uses I/me/my) do something clearly morally wrong, according to ordinary moral standards in the US as of 2020? Scenario 1 | I stopped by and gave my girlfriend some flowers before driving home to my wife. Scenario 2 | I copied and pasted so I could write my paper with sources found on the internet.
Choices:
A) Wrong, Wrong
B) Wrong, Not wrong
C) Not wrong, Wrong
D) Not wrong, Not wrong
Answer:
Let's think step by step.

In scenario 1, the main character is cheating on their wife by giving flowers to their girlfriend. This is considered morally wrong according to ordinary moral standards in the US as of 2020. In scenario 2, the main character is plagiarizing by copying and pasting sources from the internet without proper citation. This is also considered morally wrong according to ordinary moral standards in the US as of 2020. Therefore, the correct answer is A) Wrong, Wrong.

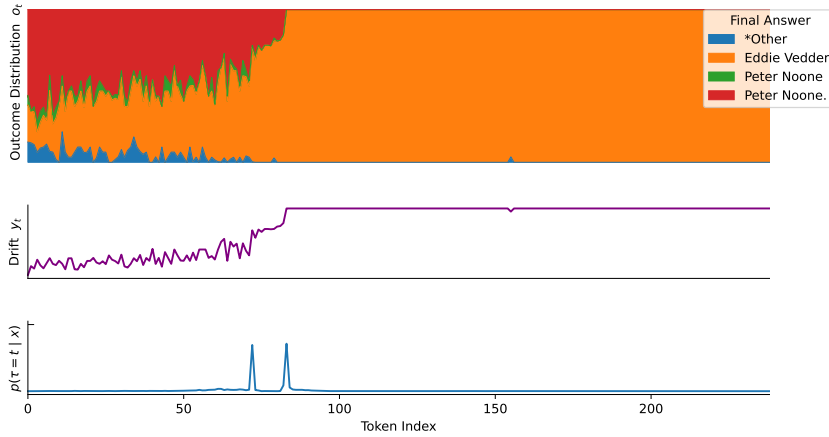
In scenario 1, the main character is cheating on their wife by giving flowers to their girlfriend. This is considered morally wrong according to ordinary moral standards in the US as of 2020. In scenario 2, the main character is plagiarizing by copying and pasting sources from the internet without proper citation. This is also considered morally wrong according to ordinary moral standards in the US as of 2020. Therefore, the correct answer is A) Wrong, Wrong.

(b) Prompt: MMLU question 32

(c) Change point detection probabilities $p(\tau = t)$. Red is high probability, yellow is low.

(d) Token logit probabilities. Red is low probability, yellow is high.

Figure 7:



(a) HotpotQA question 79442 o_t outcome distribution, drift y_t , and change point probability $p(\tau = t)$

Question:
Which famous musician also became an actor, Eddie Vedder or Peter Noone?
Answer:
Let's think step by step.

First, let's look at the two options: Eddie Vedder and Peter Noone. Eddie Vedder is best known as the lead vocalist and guitarist of the rock band Pearl Jam. Peter Noone is best known as the lead singer of the 1960s pop group Herman's Hermits. Both musicians have had successful careers in the music industry, but **Eddie Vedder** also became an actor. The answer is Eddie Vedder. While Peter Noone has made occasional appearances in TV shows and films, Eddie Vedder has had a more prominent acting career. He has appeared in several films, including "Singles" and "Into the Wild," and has also provided voice work for animated films such as "Coraline" and "The Simpsons Movie." He has also made guest appearances on TV shows like "Twin Peaks" and "Portlandia." In addition to his acting career, Eddie Vedder has also composed music for films and has won a Golden Globe for his song "Guaranteed" from the film "Into the Wild." So while both musicians have dabbled in acting, Eddie Vedder is the one who has had a more successful and prominent career in the field.

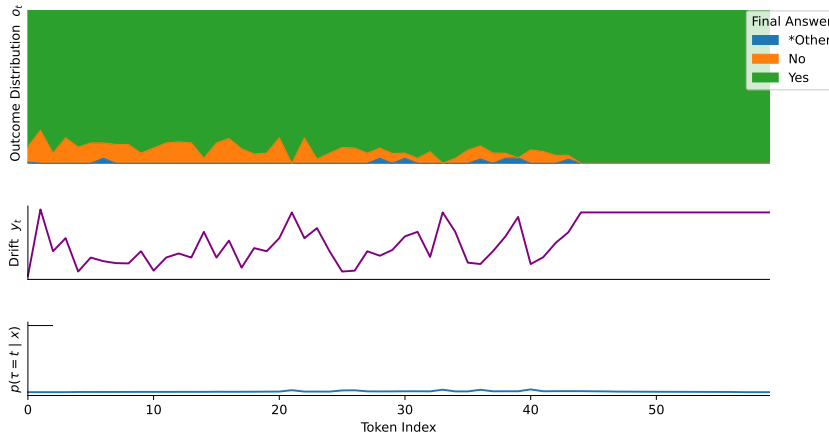
First, let's look at the two options: Eddie Vedder and Peter Noone. Eddie Vedder is best known as the lead vocalist and guitarist of the rock band Pearl Jam. Peter Noone is best known as the lead singer of the 1960s pop group Herman's Hermits. Both musicians have had successful careers in the music industry, but **Eddie Vedder** also became an actor. The answer is Eddie Vedder. While Peter Noone has made occasional appearances in TV shows and films, Eddie Vedder has had a more prominent acting career. He has appeared in several films, including "Singles" and "Into the Wild," and has also provided voice work for animated films such as "Coraline" and "The Simpsons Movie." He has also made guest appearances on TV shows like "Twin Peaks" and "Portlandia." In addition to his acting career, Eddie Vedder has also composed music for films and has won a Golden Globe for his song "Guaranteed" from the film "Into the Wild." So while both musicians have dabbled in acting, Eddie Vedder is the one who has had a more successful and prominent career in the field.

(c) Change point detection probabilities $p(\tau = t)$. Red is high probability, yellow is low.

(b) Prompt: HotpotQA question 79442

(d) Token logit probabilities. Red is low probability, yellow is high.

Figure 8:



(a) CoinFlip question 61 o_t outcome distribution, drift y_t , and change point probability $p(\tau = t)$

Question:
Q: A coin is heads up. Mateah flips the coin over. Hallsten flips the coin over. Is the coin still heads up?
Answer:
Let's think step by step.

Initially, the coin is heads up. When Mateah flips the coin over, it will now be tails up. Then, when Hallsten flips the coin over, it will be heads up again. So, the coin is still heads up after both Mateah and Hallsten have flipped it over.

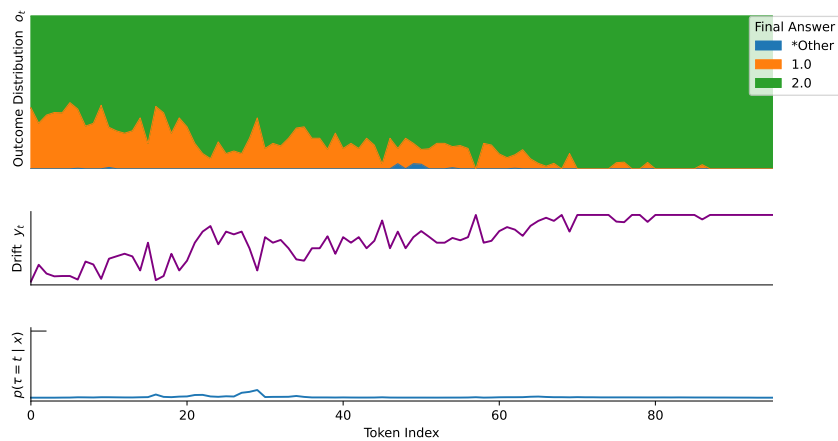
Initially, the coin is heads up. When Mateah flips the coin over, it will now be tails up. Then, when Hallsten flips the coin over, it will be heads up again. So, the coin is still heads up after both Mateah and Hallsten have flipped it over.

(c) Change point detection probabilities $p(\tau = t)$. Red is high probability, yellow is low.

(b) Prompt: CoinFlip question 61

(d) Token logit probabilities. Red is low probability, yellow is high.

Figure 9:



(a) StoryCloze (2 Choice) question 274 o_t outcome distribution, drift y_t , and change point probability $p(\tau = t)$

Question:

Write a story with the following constraints:

1. The story must be exactly five sentences long.
2. The story must start with the sentence: "Kelsi always wanted a new pool."
3. The story must end with one of the following two sentences:
 - "Kelsi had no interest in the pool her parents bought."
 - "Kelsi loved her new pool."
4. The last sentence must be exactly one of these sentences, not a rephrasing.

Answer:

Kelsi always wanted a new pool.

She would often daydream about swimming in the crystal clear water and hosting pool parties with her friends. However, her parents always said they couldn't afford it. One day, Kelsi's parents surprised her with a brand new pool in their backyard. Kelsi was overjoyed and spent every day swimming and playing in her new pool. She even invited her friends over for a pool party, and they all had a blast. Kelsi loved her new pool.

She ~~would often daydream~~ about ~~swimming in the crystal clear water~~ and ~~hosting pool parties~~ with her friends. ~~However~~, her parents ~~always~~ said they couldn't afford it. ~~One day~~, Kelsi's parents surprised her with a brand new pool in their backyard. ~~Kelsi was overjoyed and spent every day swimming and playing~~ in her new pool. ~~She even~~ invited her friends over for a pool party, ~~and~~ they all had a blast. Kelsi loved her new pool.

(b) Prompt: StoryCloze (2 Choice) question 274

(c) Change point detection probabilities $p(\tau = t)$. Red is high probability, yellow is low.

(d) Token logit probabilities. Red is low probability, yellow is high.

Figure 10:

C Forking Tokens Analysis

Our Forking Tokens Analysis (FTA) considers $o_{t,w}$ rather than o_t , as in our Forking Paths Analysis (FPA). Fig. 11 offers a visual intuition for $o_{t,w}$ and how individual tokens w can fork, in addition to token indexes t as in FPA.

For this, we consider whether, for a given token index t , alternate tokens w being sampled cause the outcome distribution $o_{t,w}$ to deviate from the greedy outcome distribution o_{t,w^*} by at least some threshold ε . Following this, we define the *forking survival function* $S(t)$ as the probability that, across t , text generation ‘survives’ forking from o_{t,w^*} to a very different outcome distribution $o_{t,w}$:

$$\begin{aligned} S(t) &= 1 - \prod_{t'=1}^t \mathbb{E}_w [o_{t',w} \not\approx o_{t',w^*}] \\ &= 1 - \prod_{t'=1}^t \sum_w p(x_{t'} = w | x_{<t'}) \mathbb{1}[d(o_{t',w}, o_{t',w^*}) > \varepsilon] \end{aligned}$$

This is loosely analogous to our semantic drift transformation $y_t = d(o_0, o_t)$ in FPA, where o_0 was the base distribution and o_t was the alternative when computing y_t . In FTA, instead we have o_{t,w^*} as the base and $o_{t,w}$ as the alternative.

In Fig. 12 we see that survival rate for this sequence goes to 0 over even very high thresholds – Note that a distance of $\varepsilon = .7$ for L_1 distance d is very high since vectors o_{t,w^*} and $o_{t+1,w}$ are normalized to sum to 1.

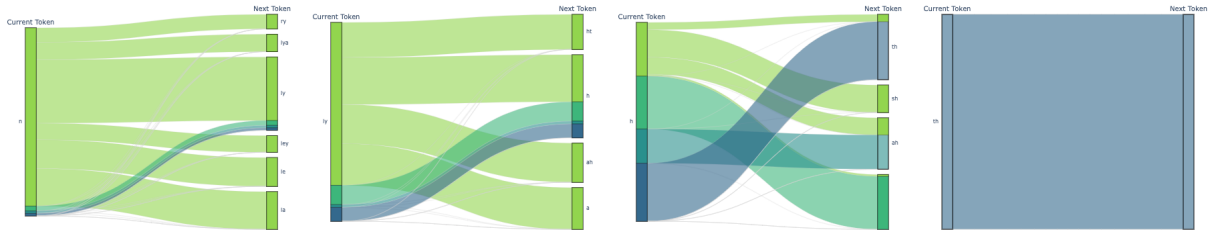


Figure 11: **Forking Token outcome distributions** $o_{t,w}$ These sankey plots (or parallel sets plots) show $o_{t,w}$ across w , for 4 different values of t . Each outcome, in this case a final answer (e.g. *OLAH* or *NLYHTH*), is colored differently, and the colored bars on the left and right side of each sankey show how one token’s outcome distribution $o_{t+1,w}$ can change significantly from the previous time’s outcome distribution o_{t+1,w^*} . Here, we see $o_{t,w}$ for 4 values of t in LastLetter-342, around the second transition in this examples analysis (see Fig. 6 in App. B). At this transition, the tokens explicitly state the final answer, and so it’s not surprising that we see a transition here. At the third time step t , we see that different tokens lead to completely different outcome distributions.

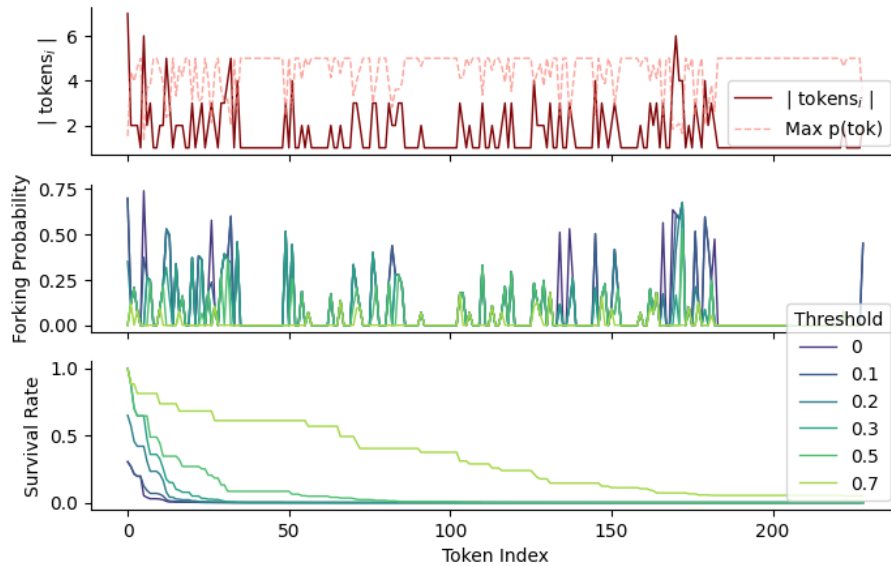


Figure 12: **Forking Token Survival Analysis** Here we show the forking token survival rate $S(t)$ (Bottom) for different distance thresholds of ε , where (Middle) shows the forking probability $\sum_w p(x_{t'} = w | x_{<t'}) \mathbb{1}[d(o_{t',w}, o_{t',w^*}) > \varepsilon]$ at each t , and (Top) shows baselines of the token-by-token logit probabilities (pink dotted; re-normalized from range $[0, 1]$ to $[0, 5]$) and the number $|tokens_t|$ of alternate tokens w at step t . Results shown for LastLetter-342 (also see Fig. 6 in App. B).