Pushing Beyond the Limitations of LLMs

Steven Rice

University of Windsor
School of Computer Science

Outline

- About me
- LLMs overview
- LLM strengths
- LLM weaknesses
- Our research here at the University of Windsor
  - Forecasting
  - Robotics
- Quick start on working with LLMs locally
- Get involved with research

University of Windsor
School of Computer Science

Outline

- **About me**
- LLMs overview
- LLM strengths
- LLM weaknesses
- Our research here at the University of Windsor
  - Forecasting
  - Robotics
- Quick start on working with LLMs locally
- Get involved with research

University of Windsor
School of Computer Science

About Me

- Master’s student at the University of Windsor
  - Vector Scholarship in Artificial Intelligence Recipient
  - Bachelor of Computer Science
  - Bachelor of Commerce
  - Minor in Mathematics
- Taught game development at the University of Detroit Mercy
- Spent some time working in industry

About Me

- Enjoy learning about and applying latest AI tech
  - A lot of research focused on robotics
- Game development and game AI

University of Windsor
School of Computer Science

Outline

- About me
- **LLMs overview**
- LLM strengths
- LLM weaknesses
- Our research here at the University of Windsor
  - Forecasting
  - Robotics
- Quick start on working with LLMs locally
- Get involved with research

University of Windsor
School of Computer Science

Large Language Models

- The latest major AI advancement
- Massive model size compared to other AI methods
  - ResNet-152 – 60.3 million parameters [1]
  - Llama 3.2 – Smallest is 1 billion parameters [2]
  - Llama 3.1 – Largest is 405 billion parameters [2]
  - Long training times and hardware requirements
- Have “emergent abilities” [3]
  - “Abilities that are not present in small models but arise in large models” [3]

Emergent Abilities

- In-context learning or “zero-shot prompting” [3]
  - Answer questions without traditional training
  - Can often perform better with “few-shot prompting”
    - Provide a few examples
- Chain-of-thought reasoning
  - Reasoning with intermediary steps to finalize an answer [4]
  - Can help solve more complex problems
- Why do these emerge?

Emergent Abilities

- Large model size and attention mechanisms
- Long prompts and even entire conversations can be fed into models
  - “Chatting” is really sending a large message of the conversation history
- Will be limited by how long the context length is
  - How many tokens can be fed in and remembered by the LLM
  - Very large in top-of-the line models – Llama models up to 128,000 tokens [2]

University of Windsor
School of Computer Science

Tokenizing

- How tokens are determined
- Used in other areas such as search engines
- It would be too much to process the input character by character
  - An algorithm will break up the input into chunks or tokens
  - 1 word ≈ 1 token
- Various strategies can be used in tokenization depending on domain

Tokenizing

- Input
  - The quick brown fox jumps over the lazy dog
- Normalization
  - **the** quick brown fox jumps over the lazy dog
- Stemming
  - The quick brown fox **jump** over the lazy dog
- Stop words removal
  - quick brown fox jumps over lazy dog
- Stemming and stop words removal are usually avoided with LLMs

University of Windsor
School of Computer Science

Outline

- About me
- LLMs overview
- **LLM strengths**
- LLM weaknesses
- Our research here at the University of Windsor
  - Forecasting
  - Robotics
- Quick start on working with LLMs locally
- Get involved with research

Strengths of LLMs

- Summarizing
- Code generation
- Creative writing

Summarizing

- No external information or knowledge needed beyond the input
  - All information essentially the LLM needs is provided by the input
- Output size < Input size
  - "Removing" information/tokens rather than expanding

University of Windsor
School of Computer Science

Code Generation

- Many LLMs trained on sources like StackOverflow
- Code is very well-structured text
    - “Easier” to understand than regular text

University of Windsor
School of Computer Science

Creative Writing

- LLMs trained on immense data sources
- Can easily pull from these when given the freedom

Outline

- About me
- LLMs overview
- LLM strengths
- **LLM weaknesses**
- Our research here at the University of Windsor
  - Forecasting
  - Robotics
- Quick start on working with LLMs locally
- Get involved with research

Weaknesses of LLMs

- Facts
- Real-time information
- Domain-specific knowledge
- Math

Facts

- Will often "hallucinate"
  - They may have been trained on facts related to a topic, but they are not exactly remembered
  - Responses which sound right may be generated that are not true upon further investigation
- Will literally make up fake references

Real-time Information

- Only know what they have been trained on or prompted with
  - If prompted with new information, this will only be a limited amount of new knowledge
- Some “chat” providers will try to search the web to provide the LLM with more information

University of Windsor
School of Computer Science

Domain-specific Knowledge

- LLMs only know what they have been trained on
- Certain domains are less common
  - Advanced topics within certain domains even more so
- Can limit the effectiveness of certain strengths such as coding
  - Certain coding languages less common
  - Asking a question in a common language but a niche package or API

Math

- Starting to improve, but still a weak point
- OpenAI’s o1 models showed great improvements [5, 6]
- Why do LLMs struggle with math?
- What can we do to improve this?

Math – Numerical Formatting

- Converting numerical values to words was found to help [7]
  - 1234 → one thousand two hundred and thirty four
  - 2% to 3% improvement in some cases [7]
- How numbers are tokenized
  - Should 1234 be treated as one token, or multiple?
  - How is it split if so?
  - If the tokenization does not split every digit, some information may be lost
  - Formatting as words ensures better tokenization

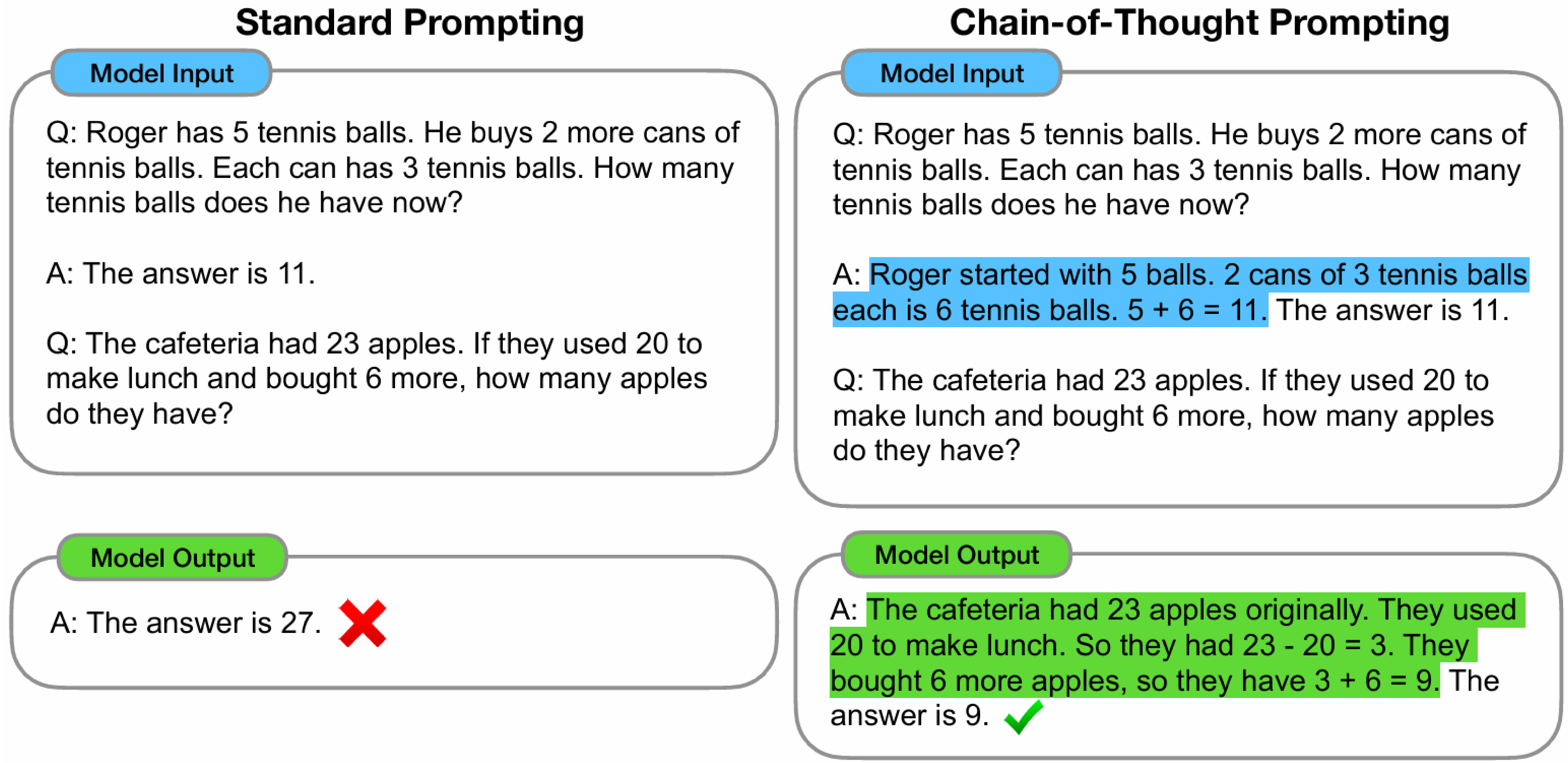
University of Windsor
School of Computer Science

Math – Show Your Work

- Prompting with examples to show steps in solving problems yielded better success [8]
- Responses showed steps as well [8]
- This is a basic example of a “chain-of-thought”

University of Windsor
School of Computer Science

Math – Show Your Work



Show your work examples [8]

Math – Chain-of-thought

- Methods to make the LLM “think” and “reflect” on its answer
- May involve a “conversation”
  - LLM essentially responding to itself
  - A user or other agent communicating with the LLM to guide it
- The o1 models use an advanced chain-of-thought method [5, 6]

Math – Multiple Attempts

- Prompt a model multiple times or different models with the same question
- The answer which occurs the most is chosen as the final answer
- “Self consistency” [9]
- Can take significantly longer

University of Windsor
School of Computer Science

Math – Leverage Strengths

- Avoid doing math with the LLM!
- Instead of focusing on solving the specific math problem, generate code that calculates the final answer [10, 11]

Outline

- About me
- LLMs overview
- LLM strengths
- LLM weaknesses
- **Our research here at the University of Windsor**
  - **Forecasting**
  - Robotics
- Quick start on working with LLMs locally
- Get involved with research

Forecasting

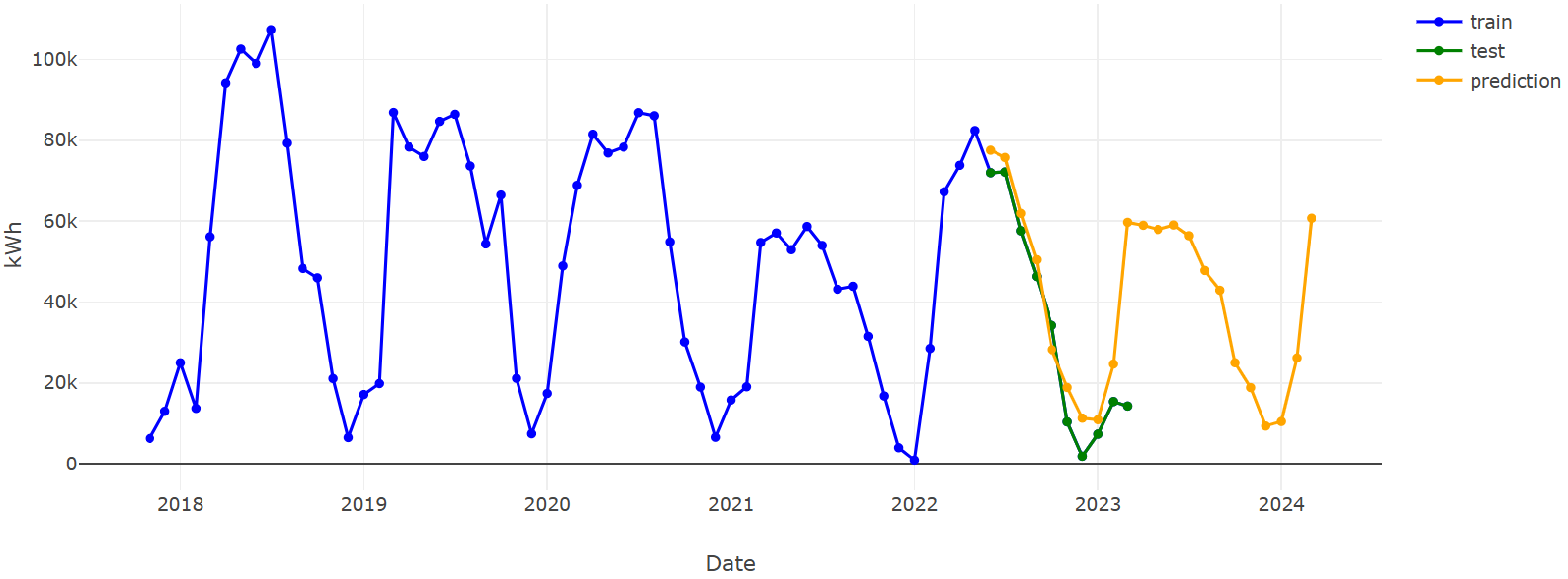
- What is required?
  - Understanding numerical data
  - May need up-to-date information
  - Potentially domain-specific knowledge
- LLMs struggle with all of these!
  - Research as shown LLMs are not good for forecasting [12]

University of Windsor
School of Computer Science

Forecasting

- Luckily, we don't have to use LLMs!
- Deep-learning methods can fit time-series data to predict the future
- Mathematical models can fit consistent data
- Where could we not use these methods?

Forecasting



Sample predictive analysis [13]

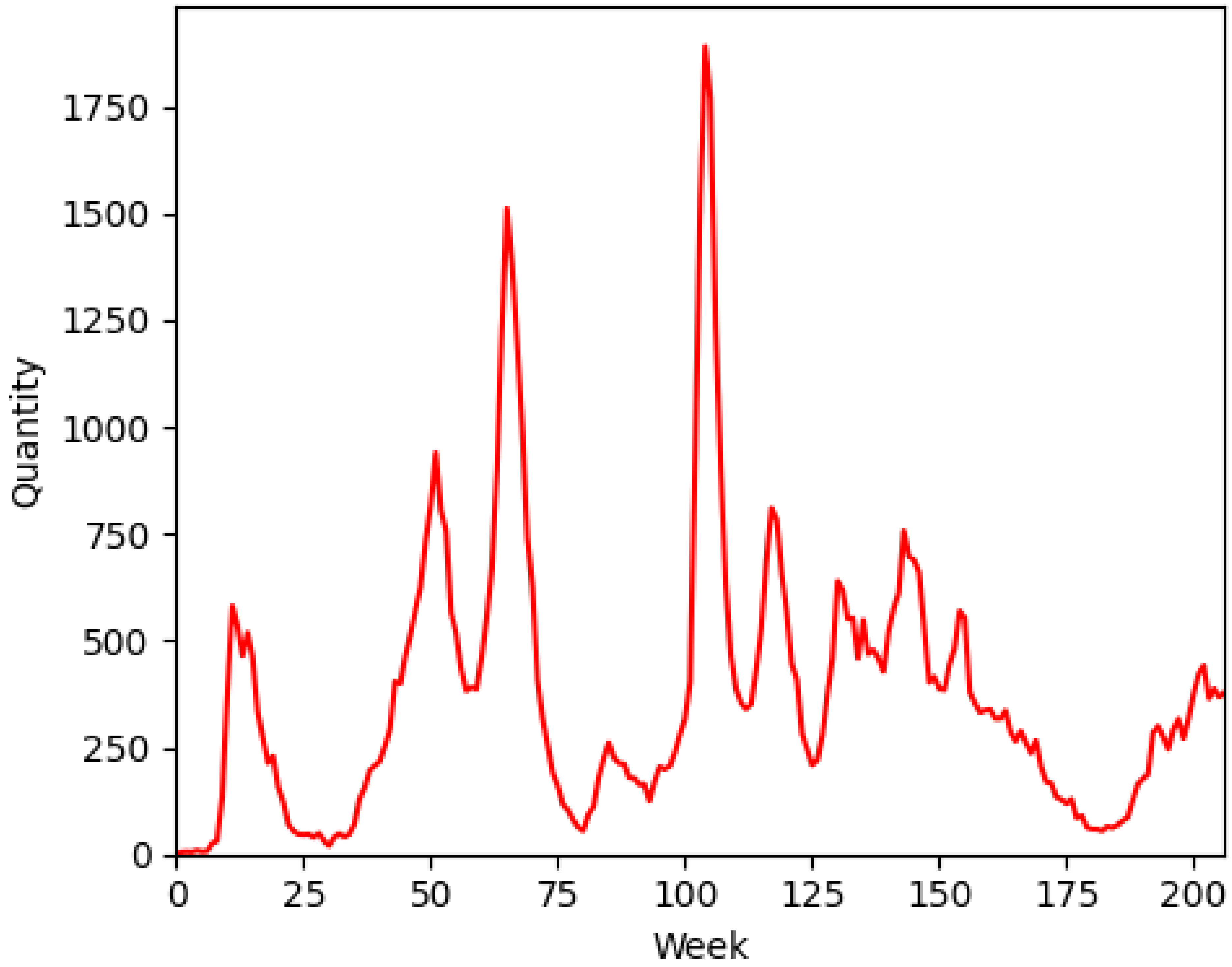
University of Windsor
School of Computer Science

Forecasting

- What if we don't have existing data?
  - How could we fit a deep learning method or create a mathematical model?
  - Are there situations like this?
- What if the data does not appear cyclical?

University of Windsor
School of Computer Science

Forecasting



University of Windsor
School of Computer Science

COVID-19 Forecasting

- No models could accurately forecast COVID-19 when the pandemic started
- Attempts to extend existing models of other diseases could not adapt for unique characteristics of COVID-19 [14]
- Deep learning could not be performed
  - Later after the start of the pandemic deep learning methods found some success [15, 16]
- What options do we have to work with?

COVID-19 Forecasting

- Simple mathematical models
  - Able to fit trends with only small amounts of data
  - ARIMA – Autoregressive integrated moving average
    - Struggles with sudden changes [17]
- Can LLMs help us here despite being considered poor for forecasting?

University of Windsor
School of Computer Science

COVID-19 Forecasting with LLMs

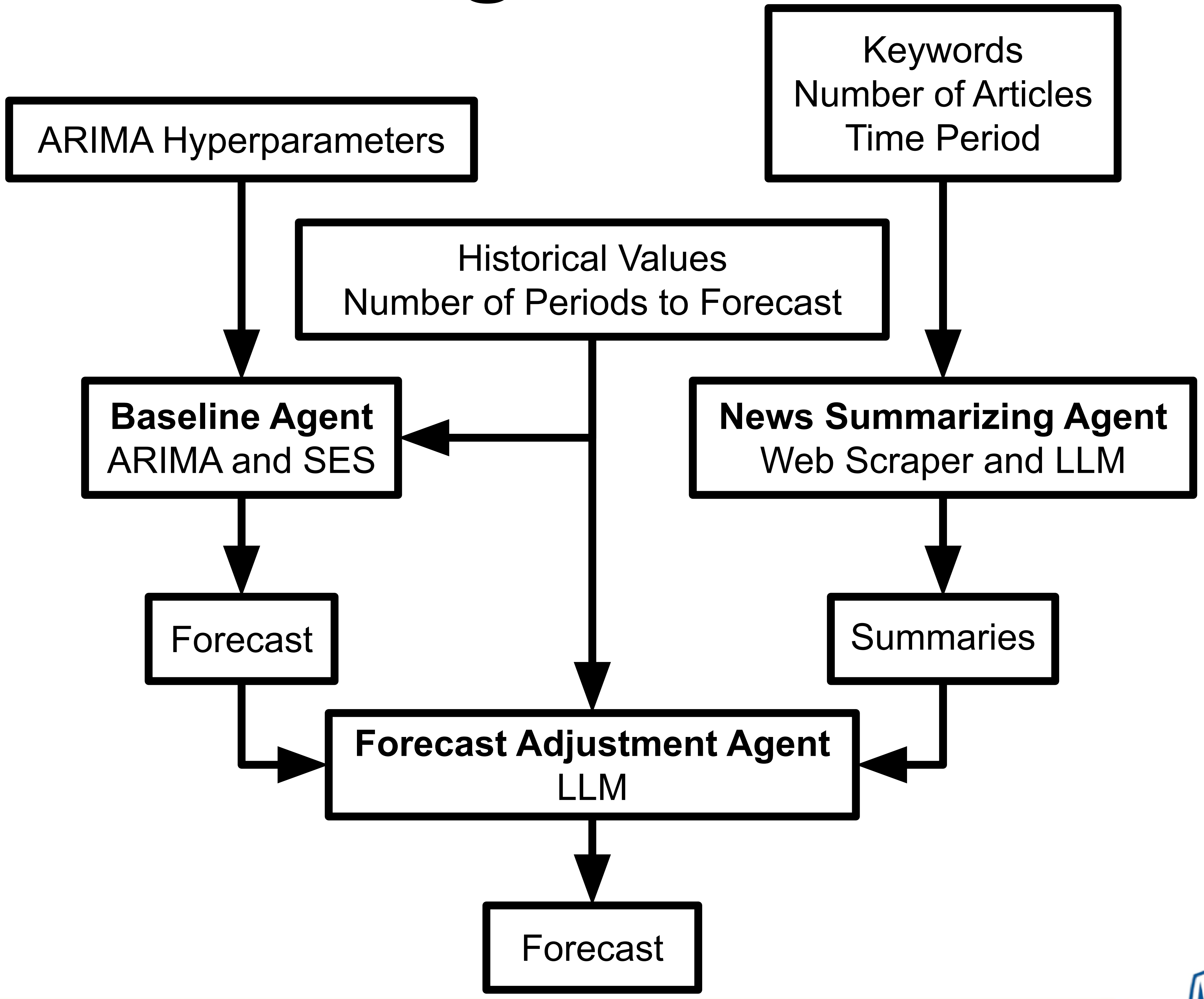
- What we need to do:
  - Make a quantitative prediction
  - Understand factors which could contribute to future COVID-19 infections
- What we can do:
  - Leverage the strengths of LLMs!
  - Quantitative prediction – no the best
  - Understand factors – potentially, if provided
    - How can we get information?

University of Windsor
School of Computer Science

COVID-19 Forecasting with LLMs

- Let's not rely on an LLM entirely!
- Quantitative prediction
  - Use a traditional forecasting model as a baseline like ARIMA and Simple Exponential Smoothing (SES)
- Need information
  - Scrape the web and use the LLM to summarize

COVID-19 Forecasting with LLMs



University of Windsor
School of Computer Science

COVID-19 Summarizing Agent Prompt

You are trying to help forecast $FORECASTING$ in $LOCATION$. Below is an article. If the article is not relevant for forecasting $FORECASTING$ in $LOCATION$, respond with "FALSE". If the article is relevant for forecasting $FORECASTING$ in $LOCATION$, respond with a brief summary highlighting values most important for forecasting $FORECASTING$ in $LOCATION$. Only state facts and keep sentences short:

$ARTICLE$

COVID-19 Adjustment Agent Prompt

You are tasked with forecasting $FORECASTING$ you predict will occur over the next $FUTURE\ PERIODS$. You must respond with a single integer and nothing else. Here are the number of $FORECASTING$ in $LOCATION$ from the previous $PAST-PERIODS$ from oldest to newest: $HISTORY_n$, $HISTORY_{n-1}$, ..., $HISTORY_2$, $HISTORY_1$. An ARIMA forecasting model has predicted that over the next $FUTURE\ PERIODS$, there will be X $FORECASTING$ in $LOCATION$. ARIMA is known to underpredict and react too slow for a surge. Using your best judgement, keep or adjust this value. Below are M news articles from the past $PERIOD$ to help guide you in making your decision.

COVID-19 Adjustment Agent Prompt

Article *I* of M
Title : $TITLE_I$
Publisher: $PUBLISHER_I$
Posted : AGE_I days ago
$SUMMARY_I$

University of Windsor
School of Computer Science

COVID-19 Forecasting with LLMs

- Applied this model on COVID-19 hospitalizations in Ontario
  - 207 weeks of data from the start of COVID-19
- Had to forecast a set number of weeks in advance
- The proposed model was compared to an ARIMA-only model

University of Windsor
School of Computer Science

COVID-19 Forecasting with LLMs

Forecast	ARIMA	ARIMA + LLMs	Improvement
1	51.69%	70.53%	18.84%
2	52.43%	67.48%	15.05%
3	50.73%	63.41%	12.68%
4	49.02%	57.84%	8.82%
5	49.26%	55.17%	5.91%
6	48.02%	53.96%	5.94%
7	48.76%	53.73%	4.98%
8	46.00%	51.00%	5.00%
9	45.23%	49.25%	4.02%
10	45.45%	47.47%	2.02%
11	47.21%	51.27%	4.06%
12	46.43%	52.04%	5.61%

Success Rate Improvement

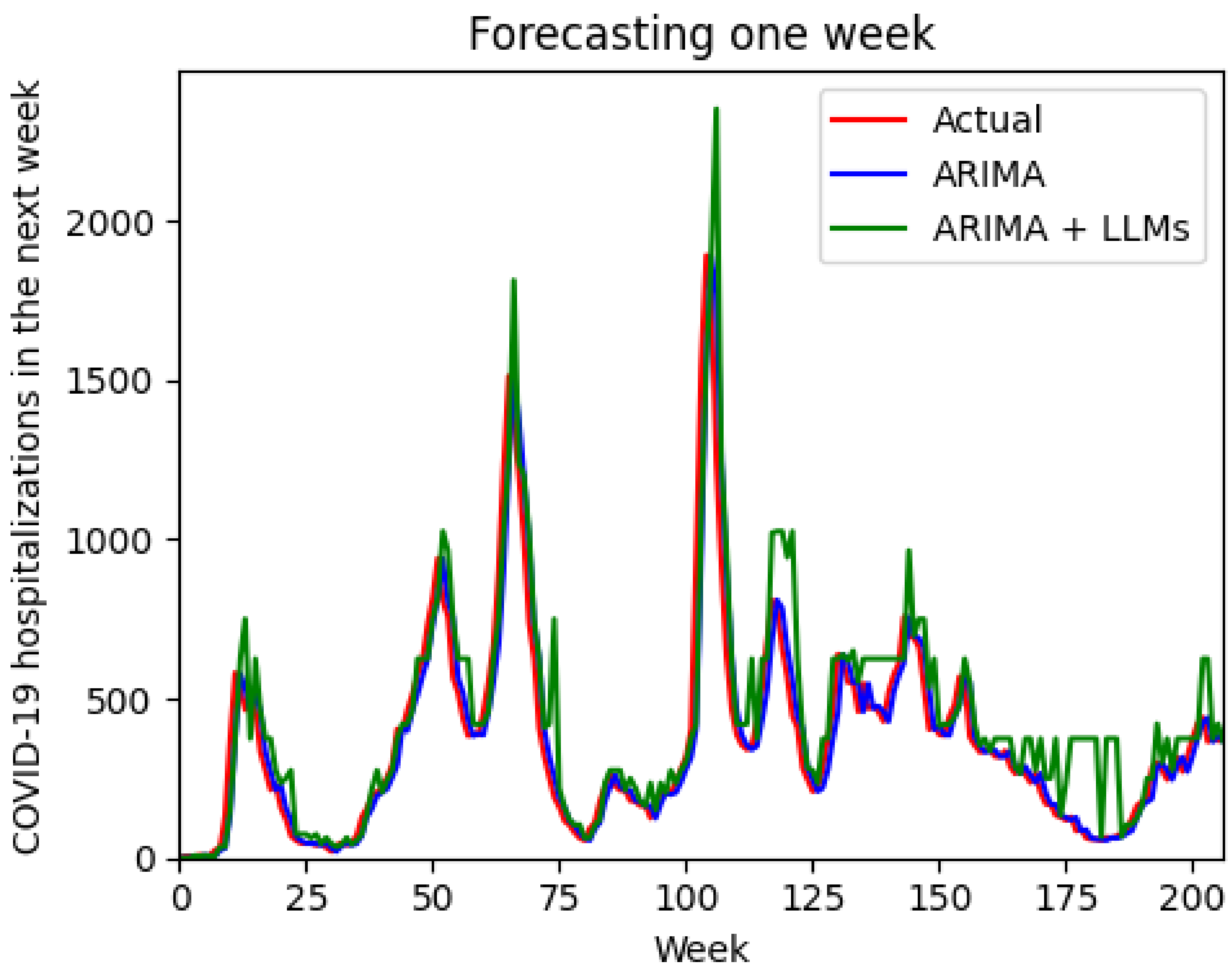
COVID-19 Forecasting with LLMs

Forecast	ARIMA	ARIMA + LLMs
1	-2.33	68.32
2	-5.88	79.77
3	-11.48	122.86
4	-18.49	145.43
5	-29.09	142.48
6	-41.23	192.81
7	-53.81	238.92
8	-65.99	182.47
9	-77.48	219.94
10	-92.34	68.69
11	-106.8	106.82
12	-118.15	210.28

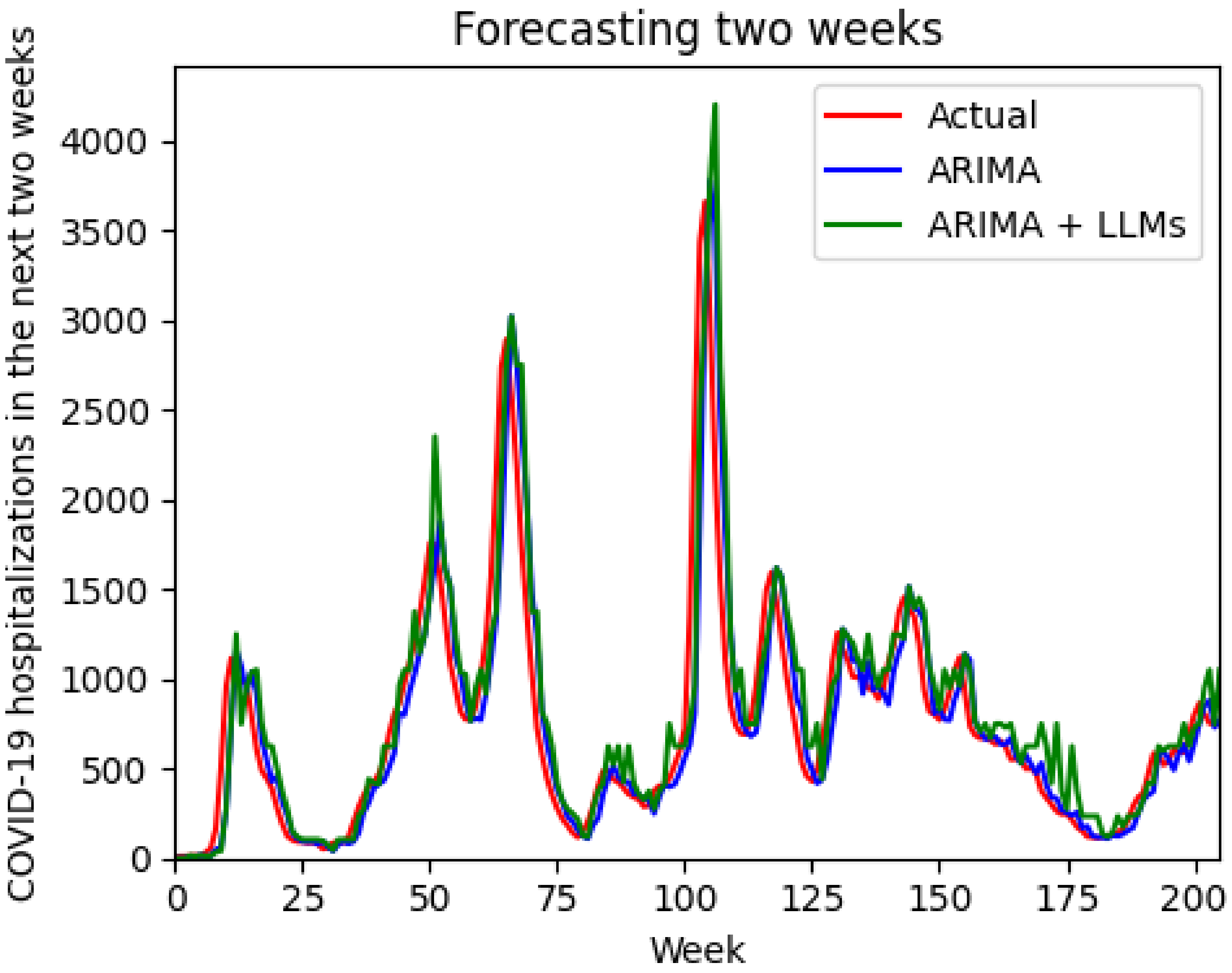
Average Difference

University of Windsor
School of Computer Science

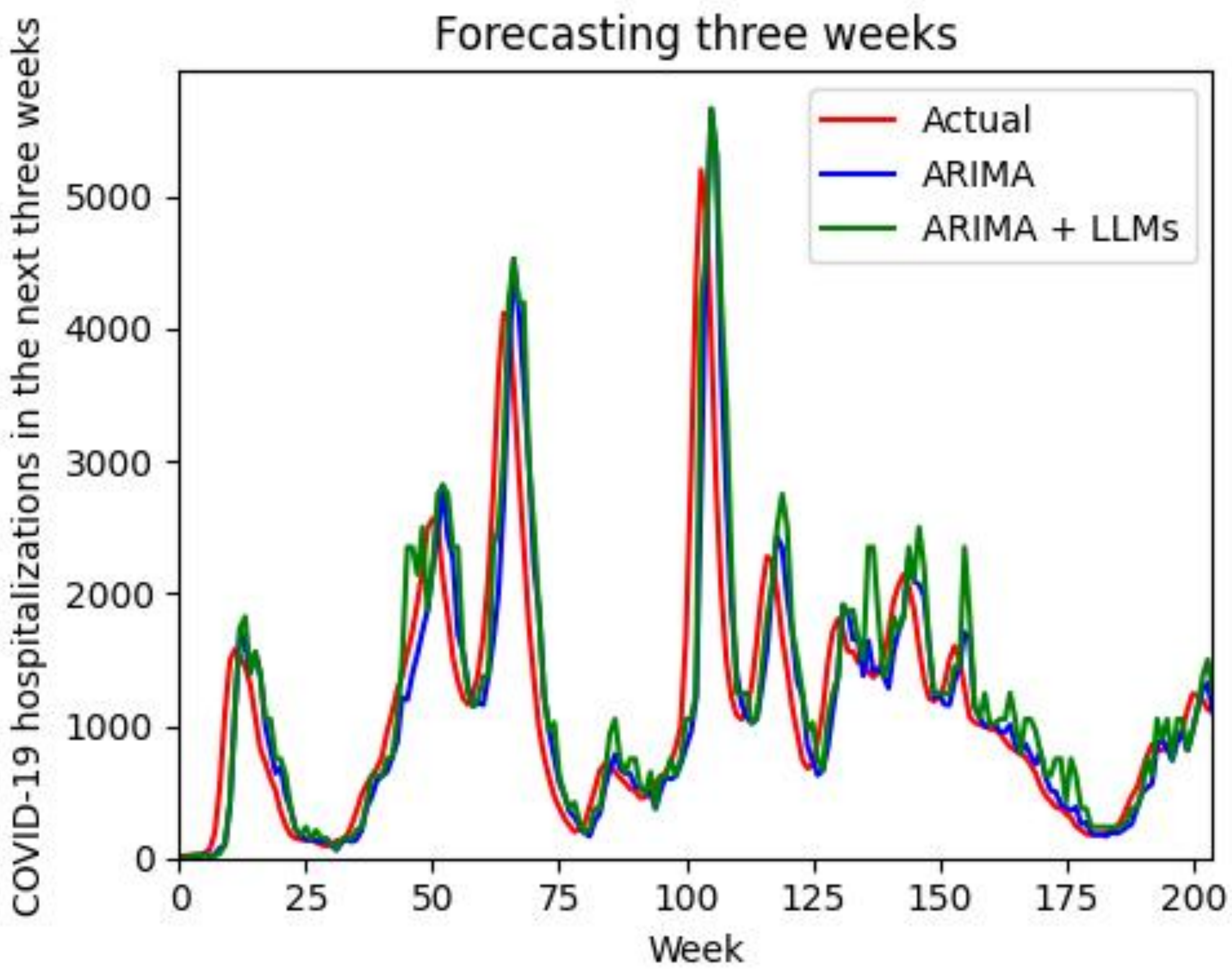
COVID-19 Forecasting with LLMs



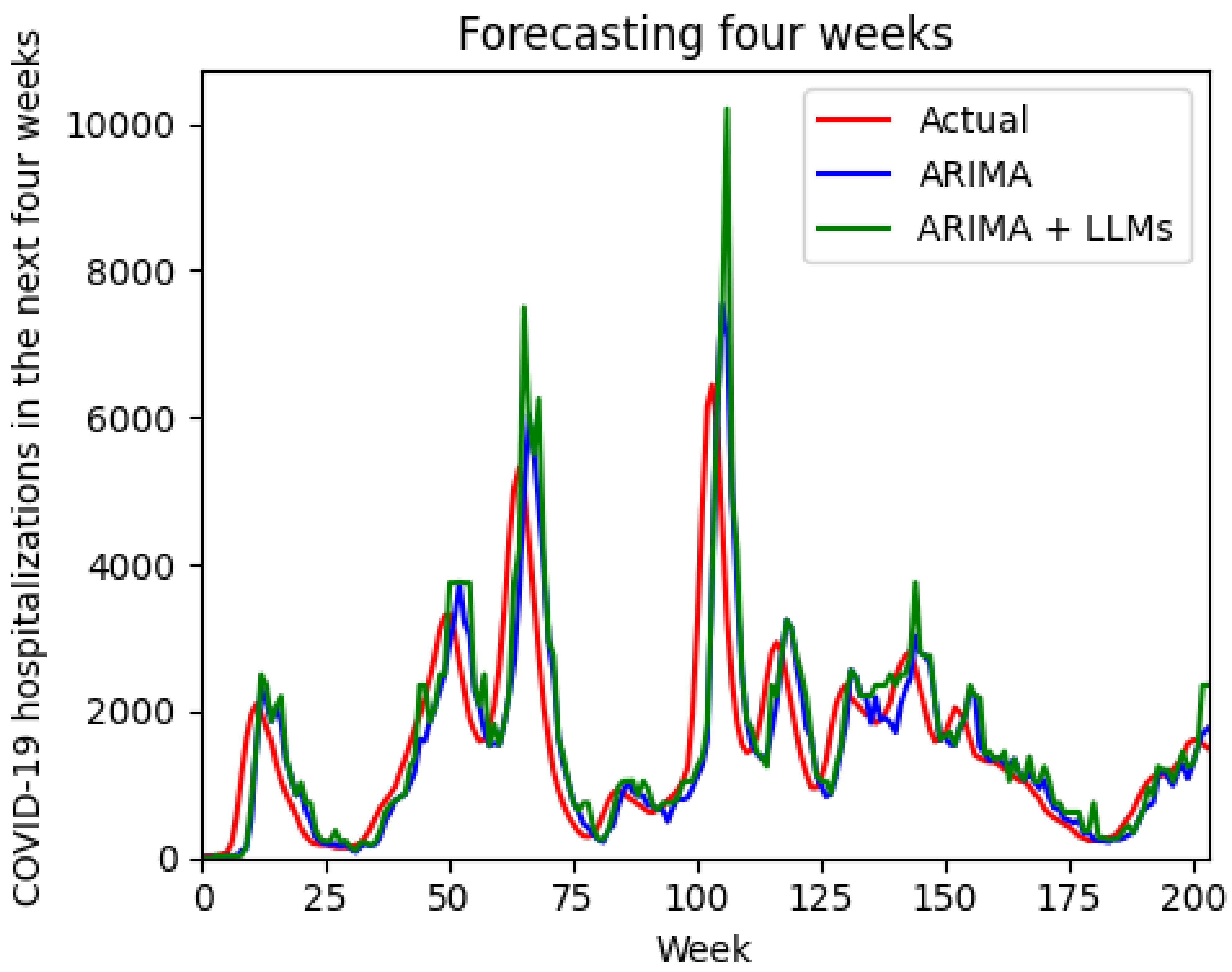
COVID-19 Forecasting with LLMs



COVID-19 Forecasting with LLMs

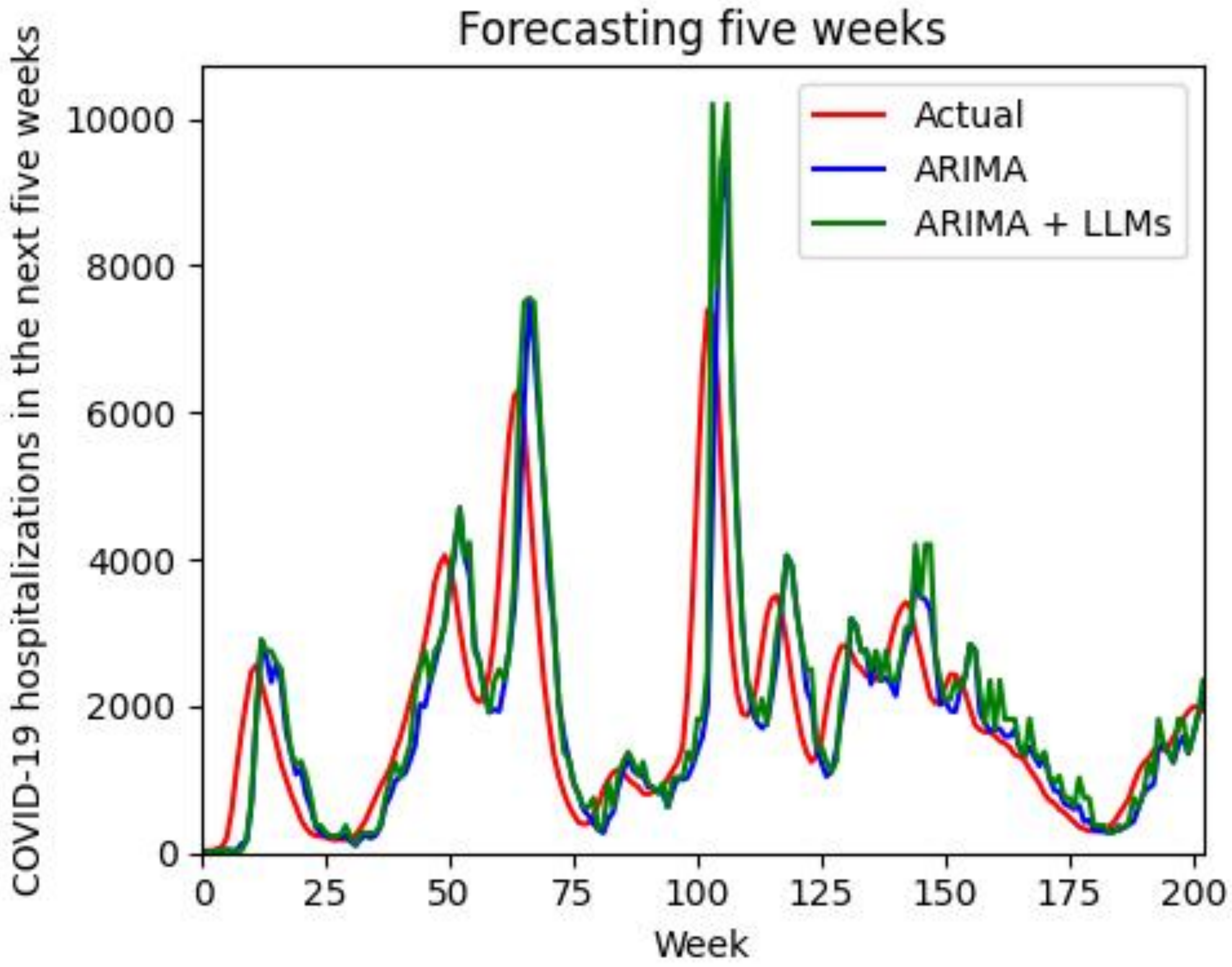


COVID-19 Forecasting with LLMs

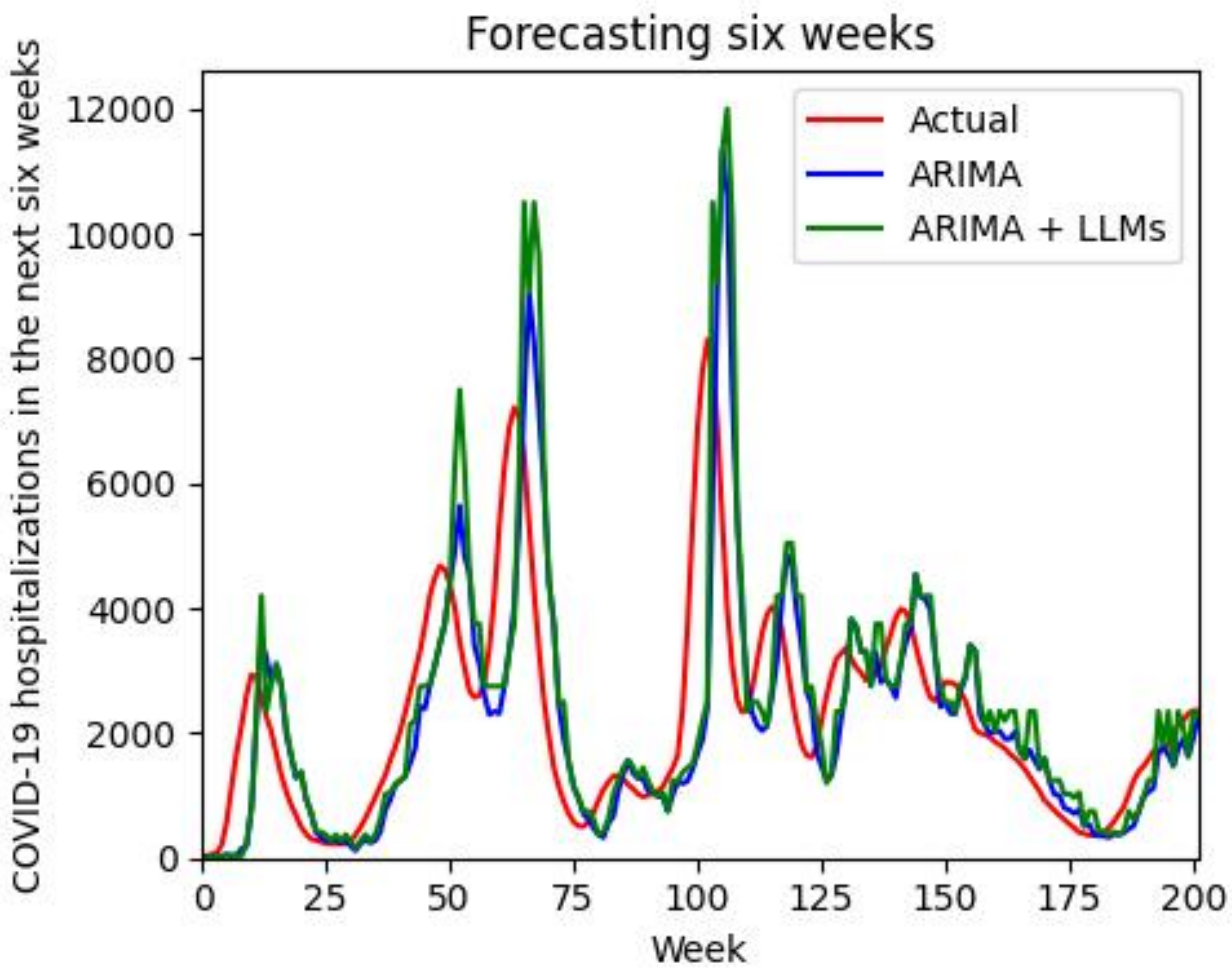


University of Windsor
School of Computer Science

COVID-19 Forecasting with LLMs

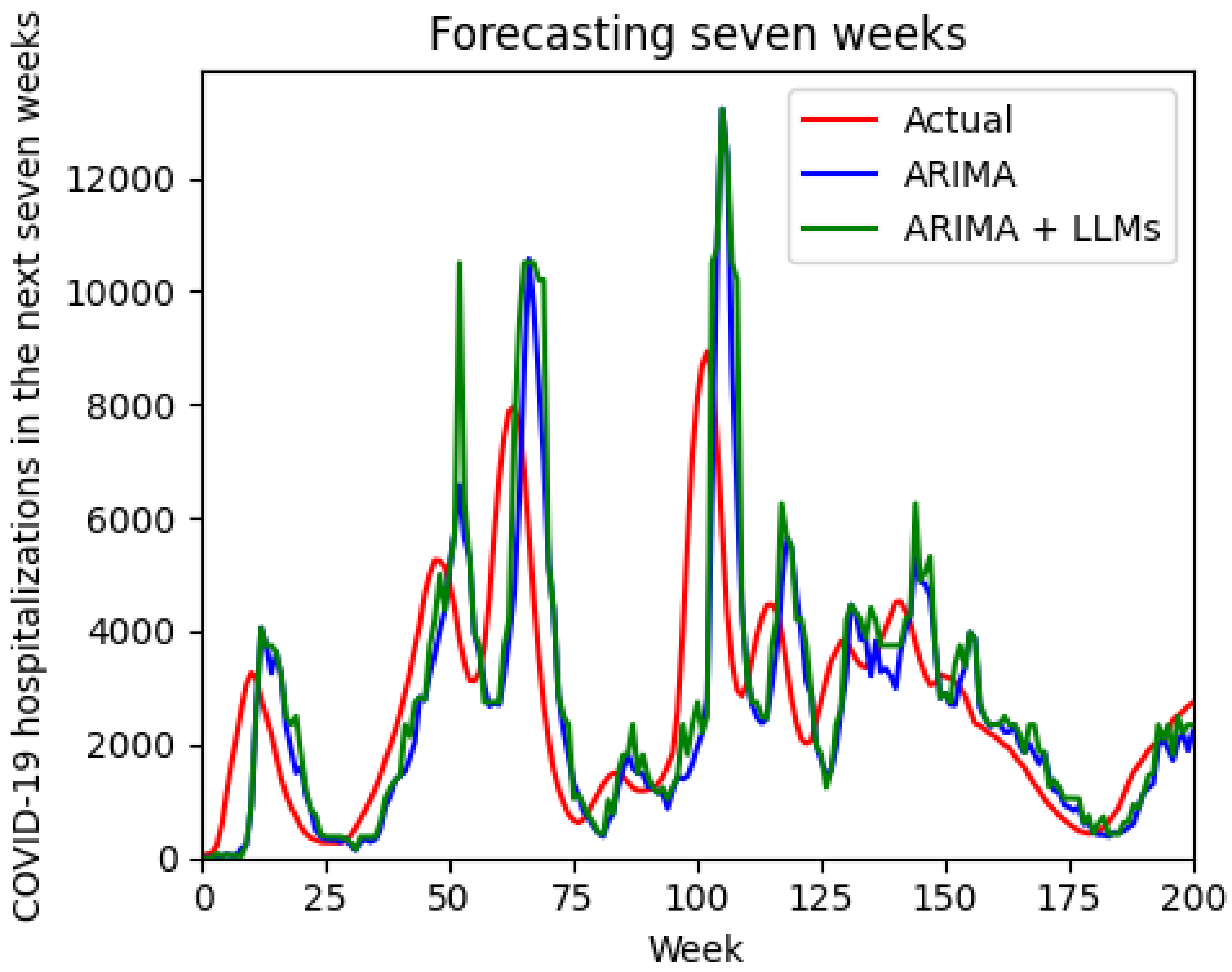


COVID-19 Forecasting with LLMs

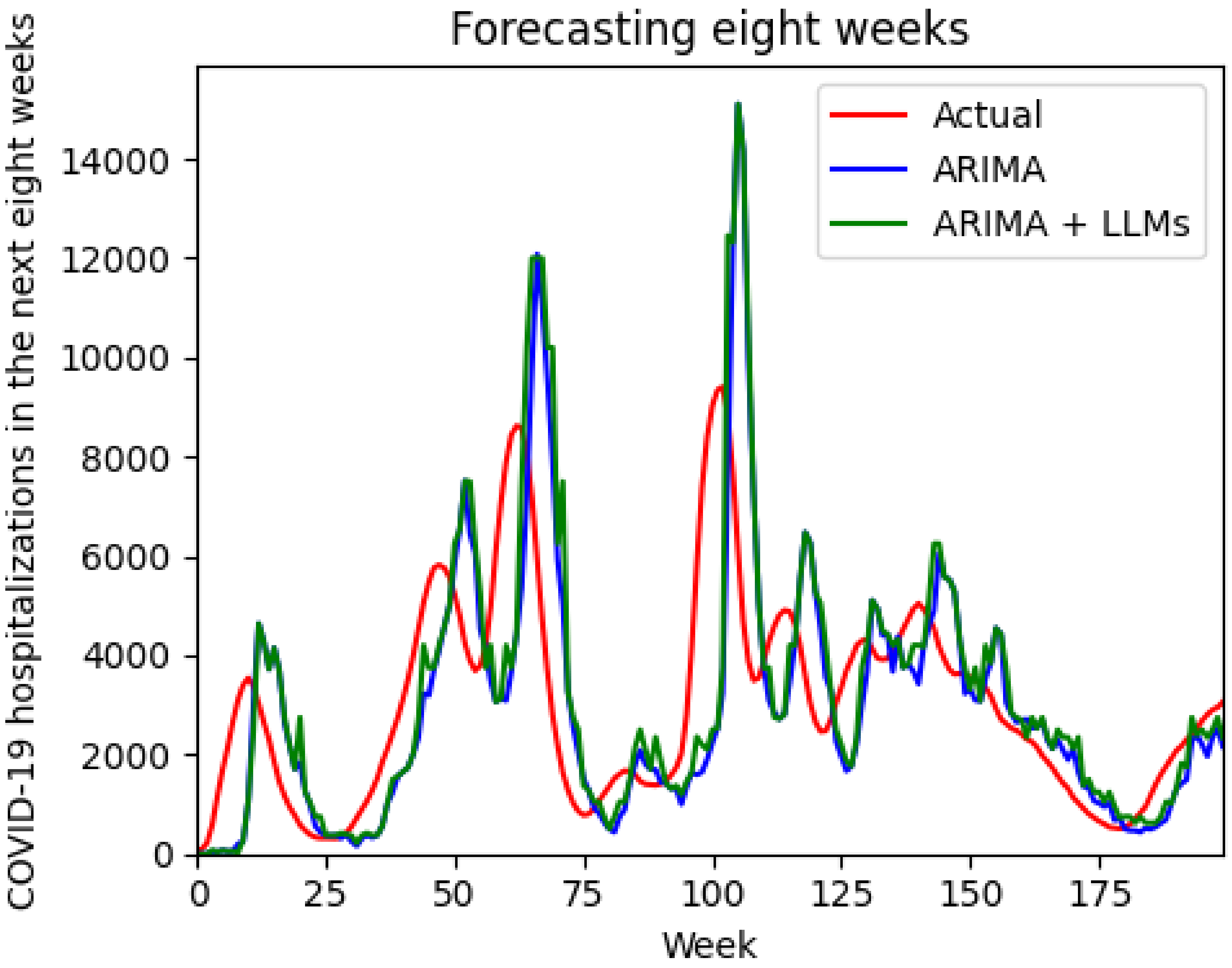


University of Windsor
School of Computer Science

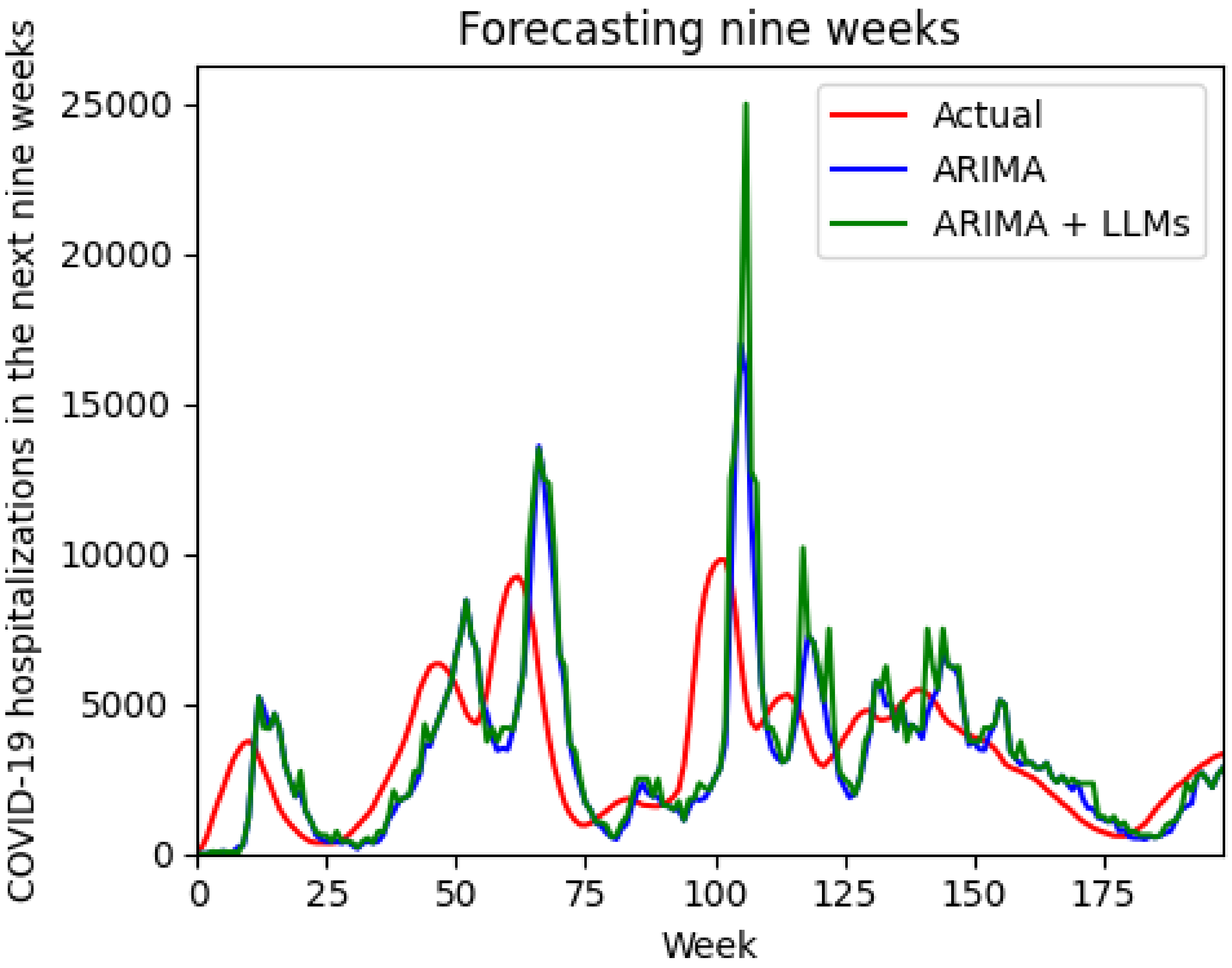
COVID-19 Forecasting with LLMs



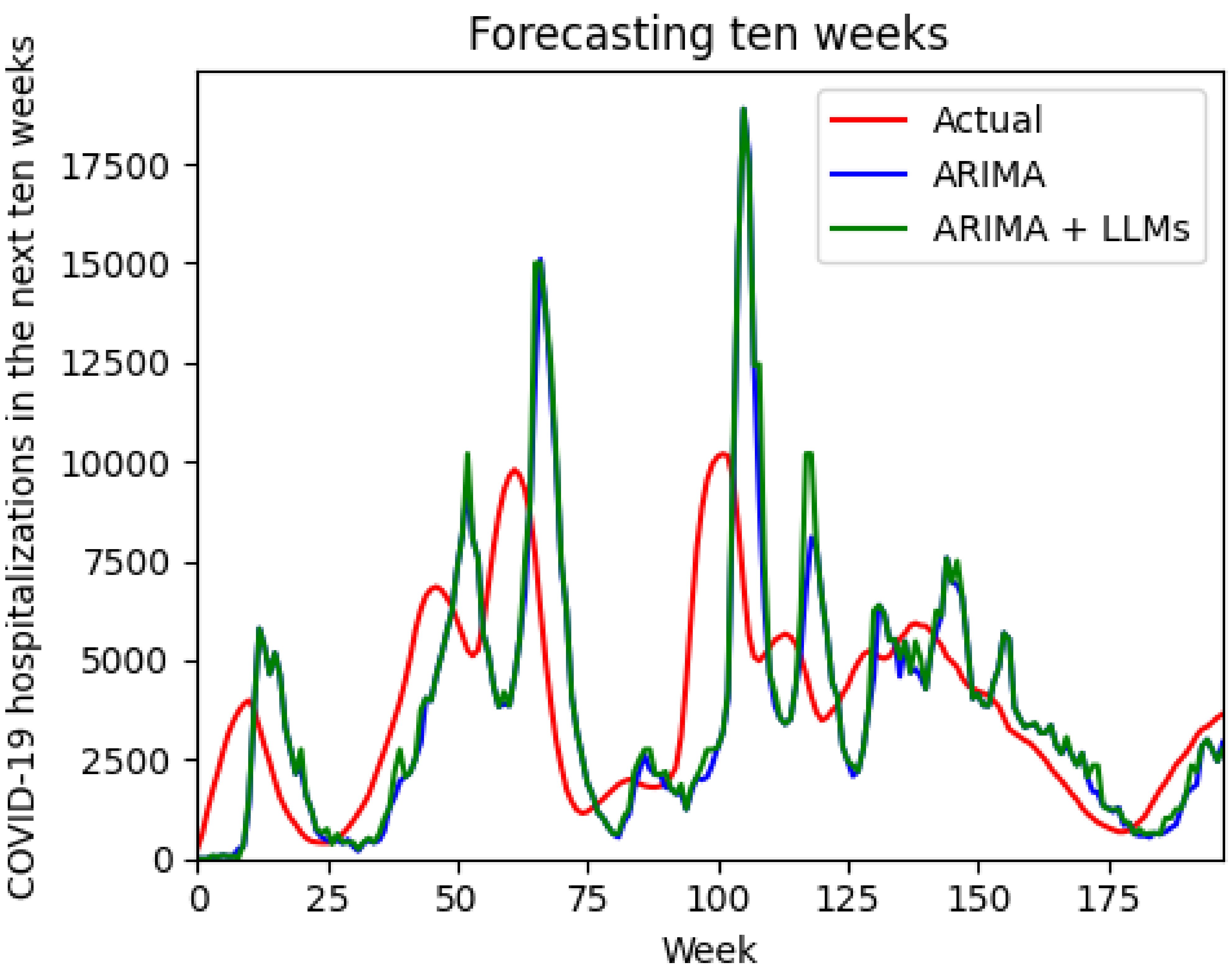
COVID-19 Forecasting with LLMs



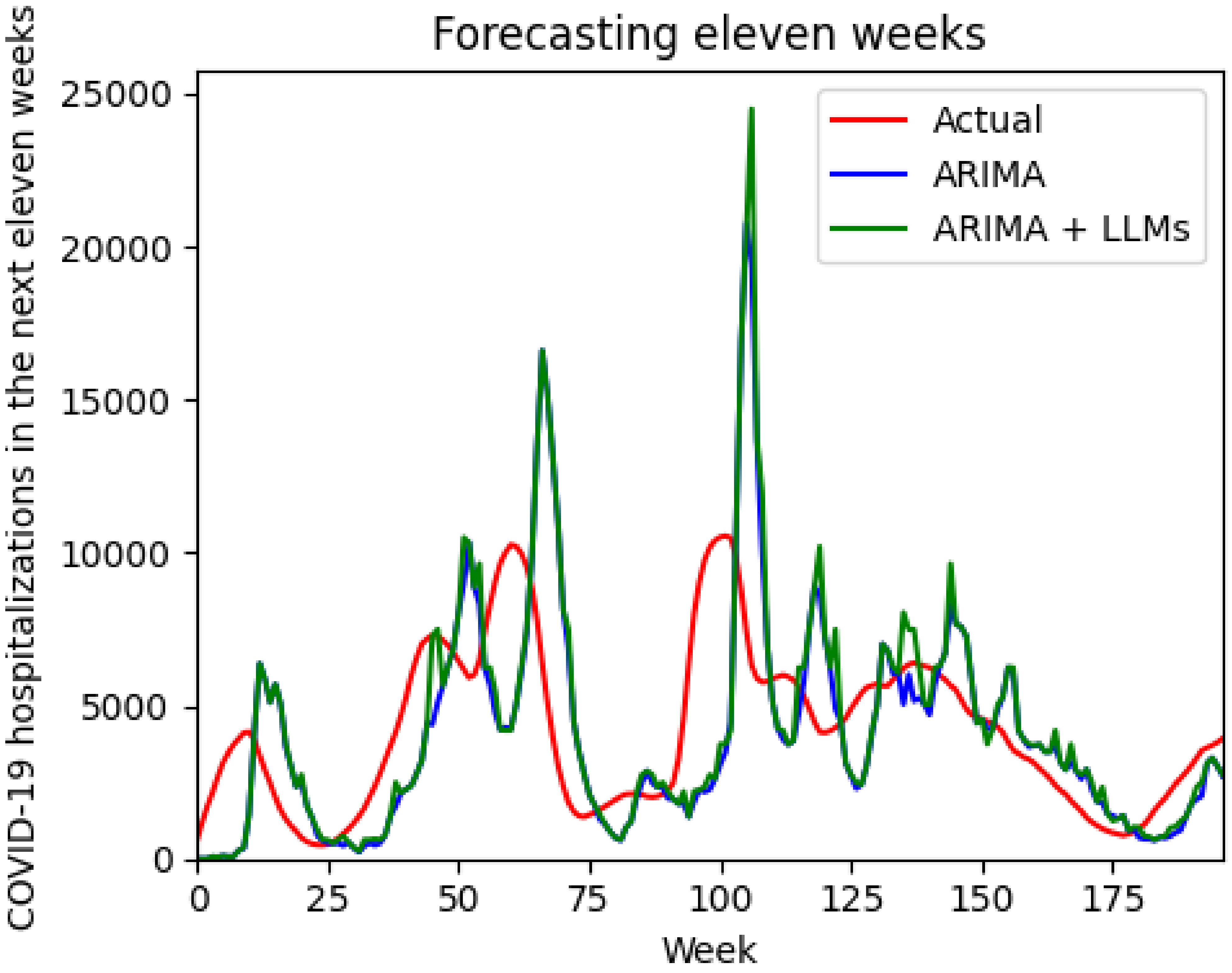
COVID-19 Forecasting with LLMs



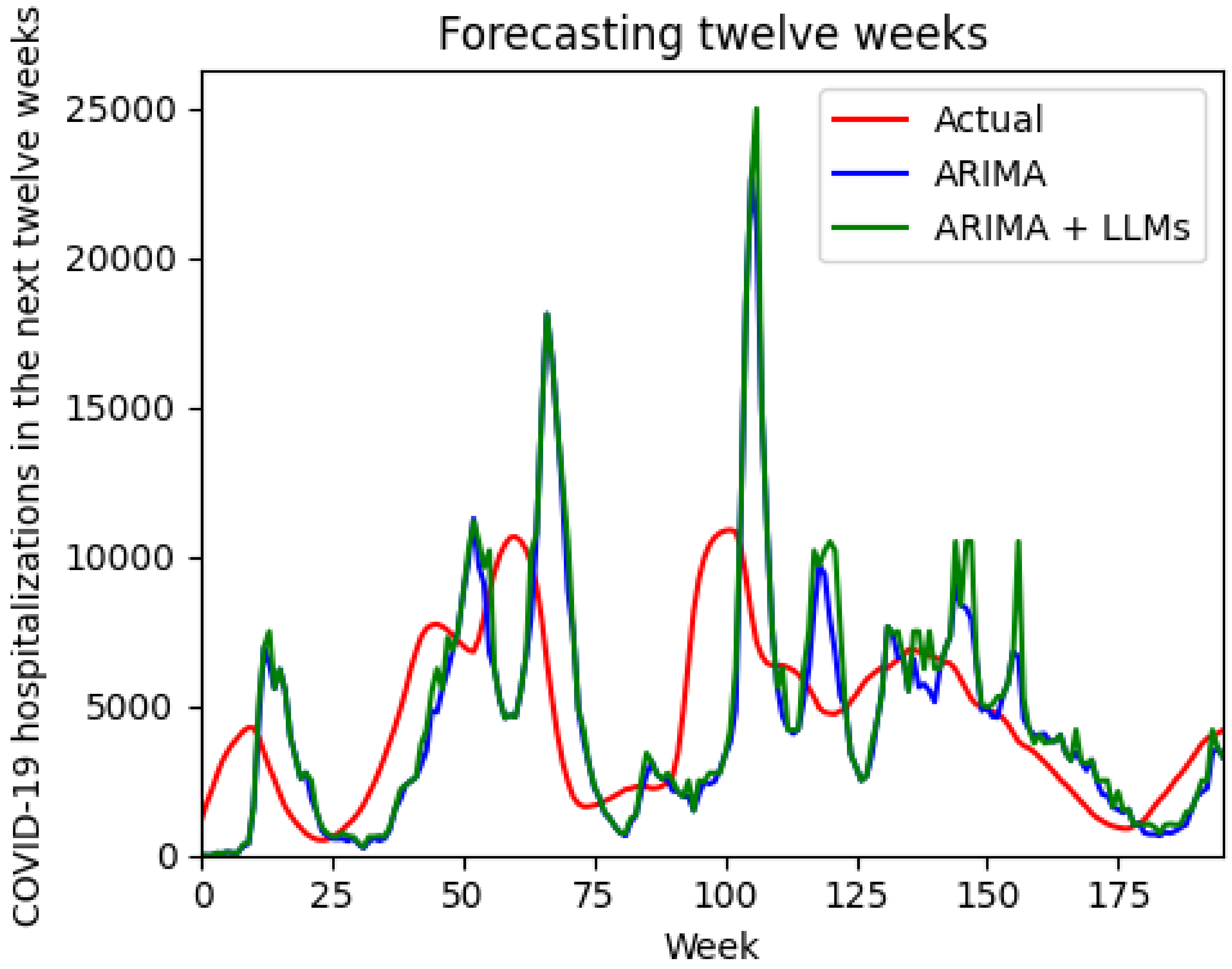
COVID-19 Forecasting with LLMs



COVID-19 Forecasting with LLMs



COVID-19 Forecasting with LLMs



University of Windsor
School of Computer Science

COVID-19 Forecasting with LLMs

- Showed overall improvements over traditional forecasting methods alone
  - First work to shown successful results with quantitative forecasting
- Achieved by leveraging strengths and minimizing where LLMs need to directly work with their weaknesses
- Tends to over predict
  - Preferable to under predicting in a field such as this
- What are some issues that still need to be overcome?
  - What milestones need to be achieved before a method like this could be safely used in the real-world?

University of Windsor
School of Computer Science

Problems to Overcome for Safe Real-world Use

- How accurate is the model?
- How confident is the model?
- How fault-proof is the model?
- Is this training biased in any way?

How accurate is the model?

- Currently over forecasting
- Preferable to under forecasting in this domain, but is it by too much?
  - Human-in-the loop to analyze forecasts?
  - How accurate do we need to be in this domain?
- For the same inputs, how varied is the output?
  - Run with different LLMs?
  - Run multiple trials?
  - How to determine what is the best?
    - Average? Min? Max?

University of Windsor
School of Computer Science

How confident is the model?

- Is the current model confident in its answers but they are wrong?
- Is it not confident?
- How can we determine this, and what could be done to overcome it?
- Potentially have the model reflect
  - List information it felt was useful and what was not
  - Look for trends in what varies the results of the LLM
- Provide more or better data sources
  - Integrated to official databases

How fault-proof is the model?

- What if the context length is exceeded?
  - Forgets its task and responds with irrelevant information
- How can we be sure the information being summarized is accurate?
  - Only take from certain sources?
  - Would this result in too little information?
  - May need a human-in-the-loop

University of Windsor
School of Computer Science

Is this training biased in any way?

- Do LLMs have a strong knowledge base of COVID-19?
  - Would this mean it is only recalling past information?
  - Would it perform worse on a new disease?
- No dates are fed into the current model
- Future works could mask the term “COVID-19” with a generic “disease” term and compare results

University of Windsor
School of Computer Science

Problems to Overcome for Safe Real-world Use

- More trials and more rigorous tests must be done
- There is a need for better forecasting metrics for these situations
  - This research can serve as a starting point, but is not ready for safe deployment into the real-world

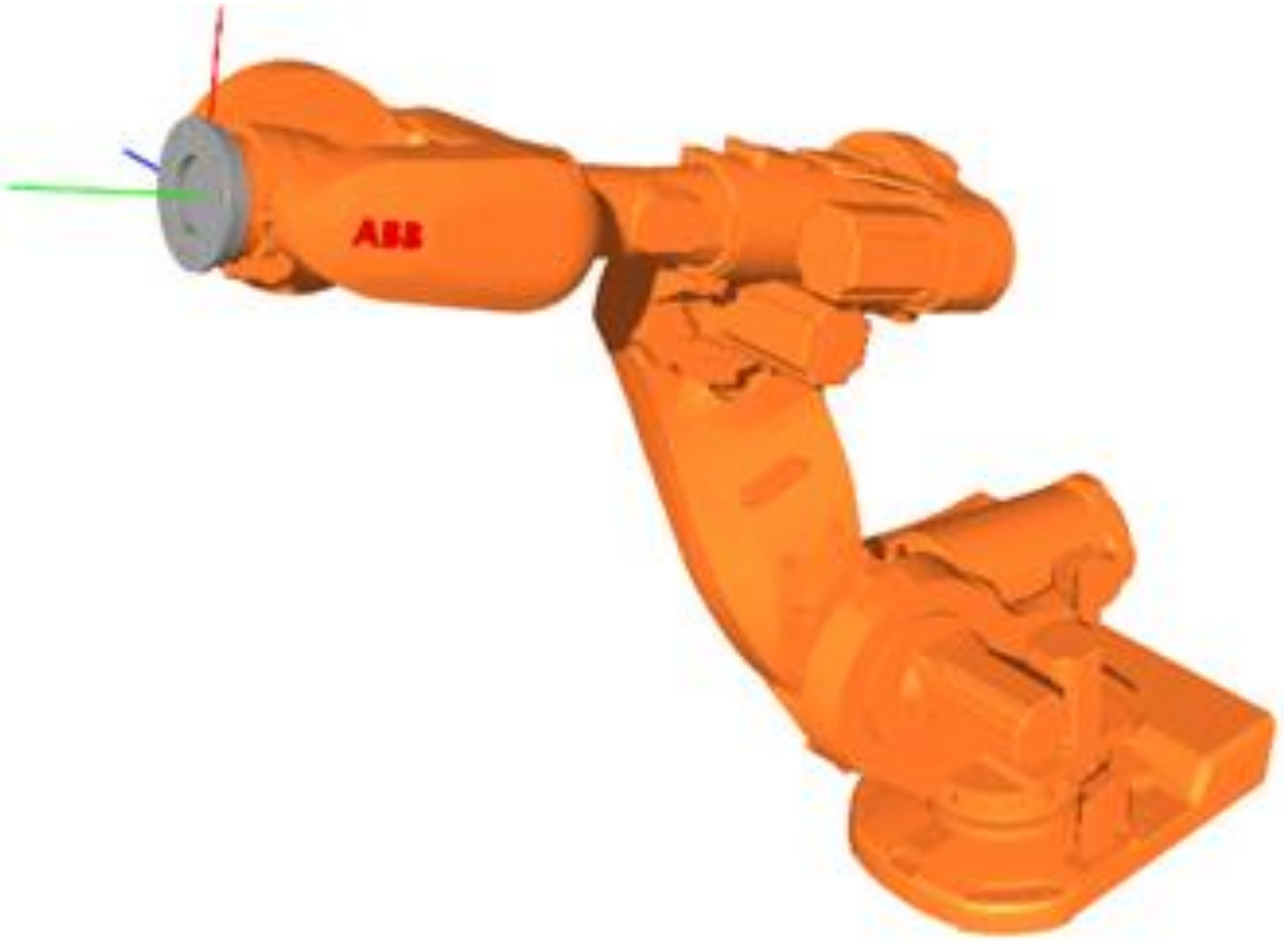
Outline

- About me
- LLMs overview
- LLM strengths
- LLM weaknesses
- **Our research here at the University of Windsor**
  - Forecasting
  - **Robotics**
- Quick start on working with LLMs locally
- Get involved with research

University of Windsor
School of Computer Science

Robotics

- Robotics are integral to all manufacturing processes
  - Typically have six axes or degrees-of-freedom
- How do robots move?
  - Inverse kinematics
  - “If I want to reach a target, where do I need to put my joints?”
- How good are inverse kinematics solving methods?



University of Windsor
School of Computer Science

Inverse Kinematics

- All methods have limitations
- Poor portability or scaling [18]
- Inconsistent accuracy, repeatability, or speeds [19, 20, 21]
- Efficient per-robot solutions often proprietary to the manufacturer
  - Works well for the real robot
  - Makes it hard to simulate for digital twins
- What does it take to solve inverse kinematics for efficient solutions?

Inverse Kinematics

- Strong math
- Ability to comprehend the kinematic structure in 3D-space
- Strong engineering skills

Solving Inverse Kinematics with LLMs

- Don’t solve directly for a particular target position
  - Generate code that can solve for any target position
- Parse and prompt with only necessary information
  - Joint positions and offsets from each other

Solving Inverse Kinematics with LLMs

1. Provide information on the kinematic chain in an initial prompt
2. Execute resulting solution on multiple test points
   1. If all tests are passed, finish
   2. Otherwise, provide feedback to the LLM to improve

- Method tested with multiple LLMs against an iterative solver from DeepMind

University of Windsor
School of Computer Science

Solving Inverse Kinematics with LLMs

```
<SUMMARY>
Produce a closed-form analytical solution for the inverse kinematics of the 3 degree-of-freedom serial
manipulator detailed in the "DETAILS" section by completing the Python function provided in the "CODE"
section. Positions and orientations are local coordinates relative to their parent link. A right-handed
coordinate system is used. The X-axis is forwards, Y-axis is left, and Z-axis is up. You may assume all
requested targets to solve the inverse kinematics for will be valid.
</SUMMARY>
<DETAILS>
Base = Position: [0, 0, 0], Orientation: [1, 0, 0, 0]
Joint 1 = Type: Revolute, Position: [0, 0, 0], Orientation: [1, 0, 0, 0], Axes: [0, 0, 1]
Joint 2 = Type: Revolute, Position: [0, 0, 0.4], Orientation: [1, 0, 0, 0], Axes: [1, 0, 0]
Joint 3 = Type: Revolute, Position: [0, 0, 0.4], Orientation: [1, 0, 0, 0], Axes: [1, 0, 0]
End Effector = Position: [0, 0, 0.4], Orientation: [1, 0, 0, 0]
</DETAILS>
<CODE>
def inverse_kinematics(p: list) -> list[float]:
    """
    Solve the inverse kinematics to reach the position in the reference frame of the chain's origin.
    :param p: The position for the end effector to reach in a Cartesian [x, y, z].
    :return: A list of the joint values in radians needed for the end effector to reach the position.
    """
</CODE>
```

Initial Prompting

University of Windsor
School of Computer Science

Solving Inverse Kinematics with LLMs

```
70% success rate solving inverse kinematics. All targets were reachable, and solutions to failures have been
solved by another inverse kinematics solver which have been provided:
Successfully reached position [-0.22133395, -0.42981172, 1.00280347]. The joints the method produced were
[-0.47554039, 0.41408481, 0.52378691].
Failed to reach position [0.11774144, -0.05791294, -0.2423271]. Instead reached position [0, 0, 1.2]. The joints
produced were [0, 0, 0]. The solution for the joints were [-2.02791513, 2.73282644, 1.22054454].
Successfully reached position [0.00785623, 0.57991123, -0.13986379]. The joints the method produced were
[-0.01354647, -2.1819756, -0.2768883].
Failed to reach position [-0.16896393, 0.00808926, 0.81507212]. Instead reached position [0, 0, 1.2]. The joints
produced were [0, 0, 0]. The solution for the joints were [-1.61863545, 1.36308653, -1.95216921].
Successfully reached position [0.31907354, 0.3135296, 0.23600698]. The joints the method produced were
[-0.79416163, -0.98936786, -1.86566995].
Successfully reached position [-0.15361328, -0.42355384, -0.18242091]. The joints the method produced were
[-0.34792351, 2.08158469, 0.80320614].
Successfully reached position [0.42372354, -0.25523604, 0.61242854]. The joints the method produced were
[1.02864, 0.33254401, 1.66527825].
Successfully reached position [-0.48461732, 0.36556683, 0.75256005]. The joints the method produced were
[0.92452262, -0.54444278, -1.00039694].
Successfully reached position [0.07368534, 0.06958759, 0.48045874]. The joints the method produced were
[-0.81399135, 0.5085246, -2.8166594].
Failed to reach position [0.48598835, 0.02773313, 0.59038701]. Instead reached position [0, 0, 1.2]. The joints
produced were [0, 0, 0]. The solution for the joints were [1.62779993, 2.05677749, -1.71756779].
```

Feedback Prompting

University of Windsor
School of Computer Science

Solving Inverse Kinematics with LLMs

Method	Success Rate (%)	Position Error (m)	Average Time (ms)	Feedbacks
o1-mini	100.00%	0.000 m	0.001 ms	0
o1-preview	100.00%	0.000 m	0.001 ms	1
GPT-4	100.00%	0.000 m	0.004 ms	1
DeepMind IK	98.80%	0.010 m	3.882 ms	-
GPT-4o	0.10%	0.502 m	0.003 ms	3

1 Degree-of-Freedom Results

University of Windsor
School of Computer Science

Solving Inverse Kinematics with LLMs

Method	Success Rate (%)	Position Error (m)	Average Time (ms)	Feedbacks
o1-mini	100.00%	0.000 m	0.004 ms	1
o1-preview	100.00%	0.000 m	0.006 ms	1
DeepMind IK	93.70%	0.046 m	4.037 ms	-
GPT-4	0%	0.010 m	0.003 ms	3

2 Degrees-of-Freedom Results

University of Windsor
School of Computer Science

Solving Inverse Kinematics with LLMs

Method	Success Rate (%)	Position Error (m)	Average Time (ms)	Feedbacks
o1-mini	100.00%	0.000 m	0.015 ms	2
DeepMind IK	73.40%	0.263 m	4.161 ms	-
o1-preview	0.10%	0.704 m	0.014 ms	3

3 Degrees-of-Freedom Results

Solving Inverse Kinematics with LLMs

Method	Success Rate (%)	Position Error (m)	Average Time (ms)	Feedbacks
DeepMind IK	83.80%	0.267 m	3.804 ms	-
o1-mini	0.00%	1.409 m	0.010 ms	3

4 Degrees-of-Freedom Results

Solving Inverse Kinematics with LLMs

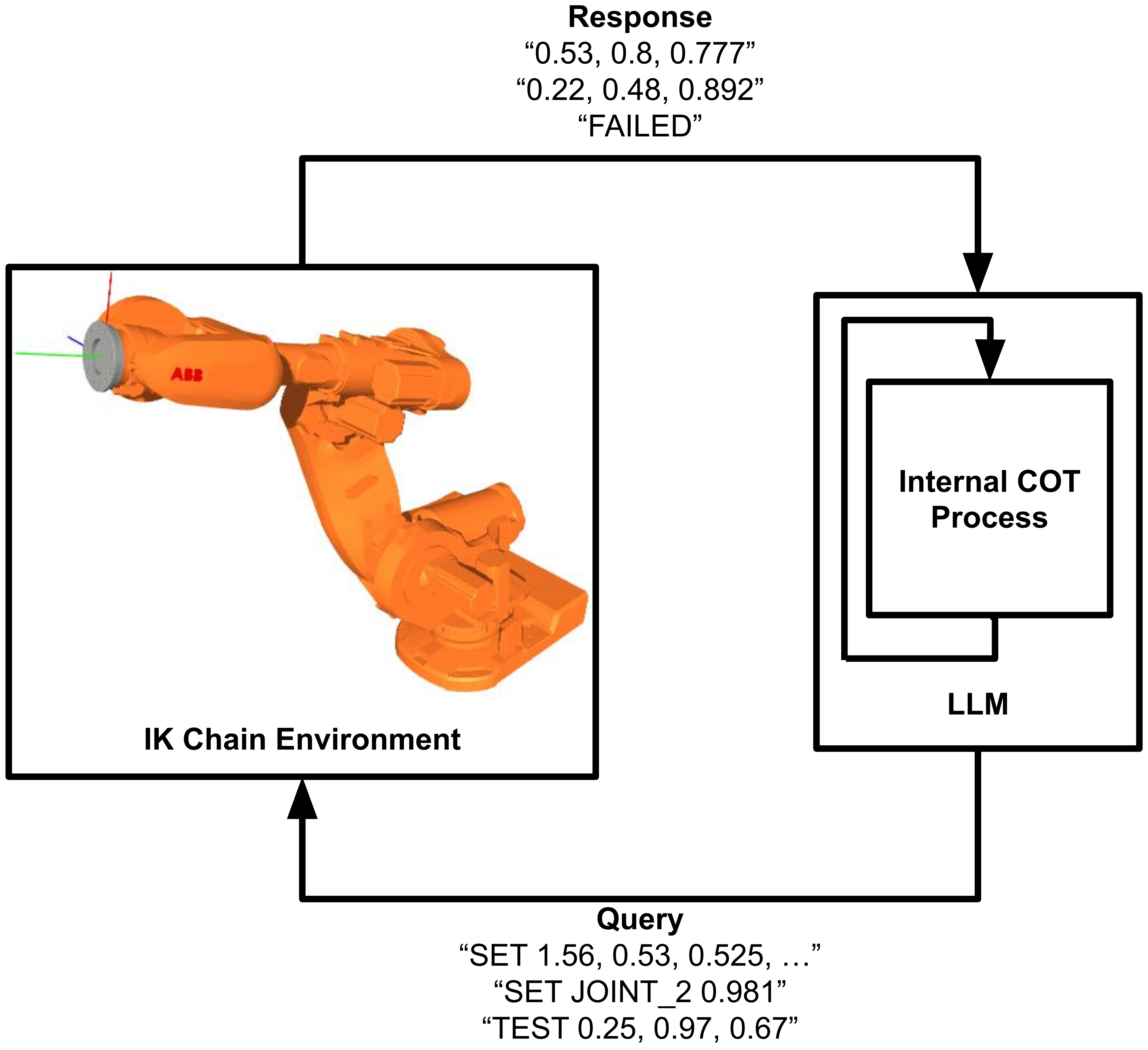
- This is the first application of LLMs towards inverse kinematics
- Successful methods were extremely efficient
  - Outperforming the DeepMind solver
- Not close to real-world use
  - Only solved three degrees-of-freedom whereas most real-world serial manipulator robots have six
- The chain-of-though capabilities of the o1 models proved effective
- How can we try to improve this?

Solving Inverse Kinematics with LLMs

- Iteratively solve
  - For a three degrees-of-freedom robot, first solve only the first joint
  - Use that result as a base for solving the second, and then that result to solve the final third joint
- Try to expand the chain-of-though process
  - Provide interactive options with the environment

University of Windsor
School of Computer Science

Solving Inverse Kinematics with LLMs



Solving Inverse Kinematics with LLMs

- Will this be enough to get to real-world use?
  - Likely not, but we need to start somewhere
  - Real robots have limits and physical limitations that are not yet being tested
    - Multiple solutions – which is the best?
- How do we verify solutions are correct?
  - Currently testing a set of random test points
    - Is this enough? How many are enough?
    - Can we conclusively say 100% of reachable points can be solved through these test cases?
  - How could we analytically prove the solutions are correct?
    - Would this require expert knowledge of kinematics to verify it?

Outline

- About me
- LLMs overview
- LLM strengths
- LLM weaknesses
- Our research here at the University of Windsor
  - Forecasting
  - Robotics
- **Quick start on working with LLMs locally**
- Get involved with research

University of Windsor
School of Computer Science

Working with LLMs locally

- Plenty of options that can run LLMs on your own computers
  - Ollama
  - vLLM
  - llama.cpp
- Most have official plugins for popular programming languages
  - Python
  - JavaScript
- Around 8 GB of RAM to run 7B models
  - Extending context lengths will require more

Working with LLMs locally

- Ollama very friendly to get started with
    1. Install Ollama
    2. Add the package
    3. Call your LLM

```python
import ollama

# Ensure we have the requested model.
ollama.pull("llama3.2")

# Generate a response from the model.
print(ollama.generate(model="llama3.2", prompt="What is five plus ten?")["response"])
```

```
5 + 10 = 15.
```


University of Windsor
School of Computer Science

Working with LLMs locally

- Create a chat system like ChatGPT

```python
import ollama

ollama.pull("llama3.2")

# Hold past messages.
messages = []
while True:
    # Get user input.
    message = input("Enter your message: ")
    messages.append({"role": "user", "content": message})

    # Get response from the LLM.
    response = ollama.chat(model="llama3.2", messages=messages)["message"]["content"]
    print(f"Response: {response}")

    # Add the response to the chat history.
    messages.append({"role": "assistant", "content": response})
```

```
Enter your message: What is five plus ten?
Response: 5 + 10 = 15.
Enter your message: Are you sure?
Response: Yes, I'm sure. The correct calculation is:

5 + 10 = 15.
Enter your message: Multiply that by two.
Response: Multiplying 15 by 2 gives:

15 × 2 = 30.
```


University of Windsor School of Computer Science

Outline

- About me
- LLMs overview
- LLM strengths
- LLM weaknesses
- Our research here at the University of Windsor
  - Forecasting
  - Robotics
- Quick start on working with LLMs locally
- **Get involved with research**

University of Windsor
School of Computer Science

Get Involved with Research

- Take advantage of the experts around you!
  - Guide and verify your ideas and give you new ones
- Most professors open to hearing your cool ideas and will support you
  - Never too early or too late to start
- University of Windsor CS undergraduates – take COMP-4960
  - One-on-one research for credit
  - Can publish a paper out of it
  - Looks great for CVs both if you decide to look for a job or continue in school

Get Involved with Research

- Scholarships favor publications
  - Can get more than enough to completely cover a masters degree if you continue your education
  - Scholarship opportunities exclusively for AI
- Vector Scholarship in Artificial Intelligence – $17,500
- Ontario Graduate Scholarship – $15,000
- NSERC Graduate Scholarships – $27,000

References

1. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
2. Meta, “Llama”, llama.com.
3. J. Wei et al., “Emergent Abilities of Large Language Models” arXiv:2206.07682 [cs.CL], June 2022, doi: 10.48550/arXiv.2206.07682.
4. J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” arXiv:2201.11903 [cs.CL], January 2022, doi: 10.48550/arXiv.2201.11903.

References

5. OpenAI, “Learning to Reason with LLMs,” September 2024
6. OpenAI, “OpenAI o1-mini,” September 2024.
7. J. An, J. Lee, and G. Gweon, “Does ChatGPT Comprehend Place Value in Numbers When Solving Math Word Problems?,” in AIED2023 Workshop: Towards the Future of AI-Augmented Human Tutoring in Math Learning, July 2023
8. J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi,Q. Le, and D. Zhou, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” 2023.

University of Windsor
School of Computer Science

References

9. X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, “Self-Consistency Improves Chain of Thought Reasoning in Language Models,” 2023.
10. A. Zhou, K. Wang, Z. Lu, W. Shi, S. Luo, Z. Qin, S. Lu, A. Jia, L. Song, M. Zhan, and H. Li, “Solving Challenging Math Word Problems Using GPT-4 Code Interpreter with Code-based Self-Verification,” 2023
11. W. Chen, X. Ma, X. Wang, and W. W. Cohen, “Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks,” 2023.

References

12. M. Tan, M. A. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen, “Are language models actually useful for time series forecasting?,” 2024.
13. J. Youssouf, “ts_forecast_demo,” github.com/JYoussouf/ts_forecast_demo.
14. K. Barrett, Y. A. Khan, S. Mac, R. Ximenes, D. M. Naimark, and B. Sander, “Estimation of COVID-19–induced depletion of hospital resources in Ontario, Canada,” CMAJ, vol. 192, no. 24, pp. E640–E646, 2020.

References

15. J. P. Ioannidis, S. Cripps, and M. A. Tanner, “Forecasting for COVID-19 has failed,” International Journal of Forecasting, vol. 38, no. 2, pp. 423–438, 2022.
16. A. L. Bertozzi1, E. Franco, G. Mohler, M. B. Short, and D. Sledge, “The challenges of modeling and forecasting the spread of COVID-19,” Applied Mathematics, vol. 117,pp. 16732–16738, July 2020.
17. I. Rahimi, F. Chen, and A. H. Gandomi, “A review on COVID-19 forecasting models,” Neural Computing and Applications, vol. 35, pp. 23671–23681, February 2021.

References

18. R. Diankov, Automated Construction of Robotic Manipulation Programs. PhD thesis, The Robotics Institute Carnegie Mellon University, August 2010.
19. S. R. Buss, “Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods,” Department of Mathematics, University of California, October 2009.
20. S. Starke, Bio IK: A Memetic Evolutionary Algorithm for Generic Multi-Objective Inverse Kinematics. PhD thesis, University of Hamburg, Faculty of Mathematics, Informatics and Natural Sciences, Department of Informatics, August 2020.

References

21. S. Rice, A. Azab, and S. Saad, “Fusion IK: Solving inverse kinematics using a hybridized deep learning and evolutionary approach”, Manufacturing Letters, vol. 41, pp. 9–18, 2024.

Thank you for listening!

StevenRice.ca

Contact@StevenRice.ca

StevenRice99

StevenRice99

University of Windsor
School of Computer Science