

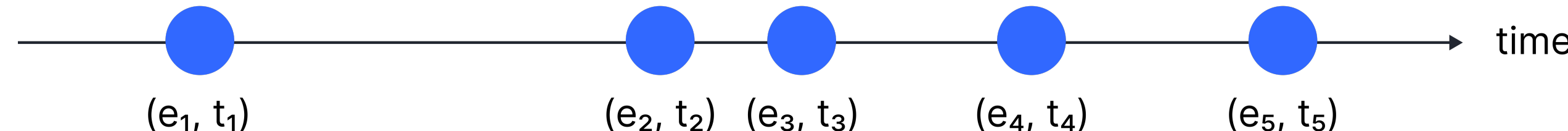
# LAST SToP For Modeling Asynchronous Time Series

Shubham Gupta, Thibaut Durand, Graham Taylor, Lilian W. Białokozowicz

**TLDR: LAST SToP is an efficient method to adapt LLMs for asynchronous time series while preserving semantic information through language.**

## Asynchronous Time Series

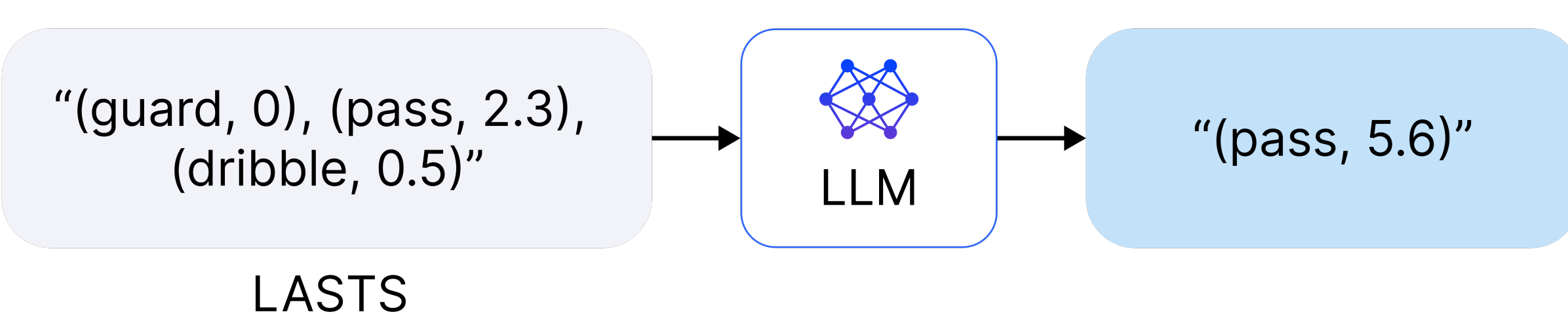
- Sequence of events that occur at irregular time intervals
- Each event is represented by a pair (event type, time)
  - Event type is a string representation of the event



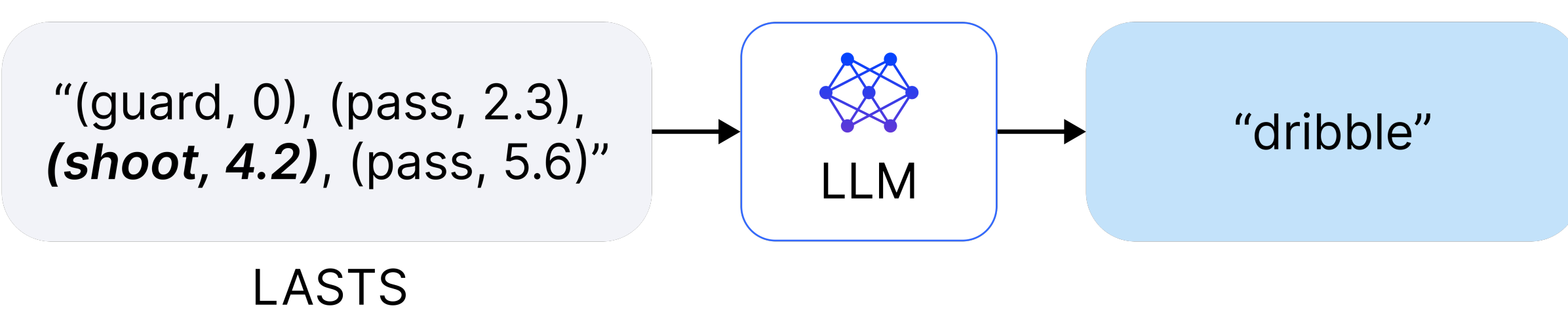
- Examples: events in a basketball game, financial transactions, social media activities

## Asynchronous Time Series Tasks

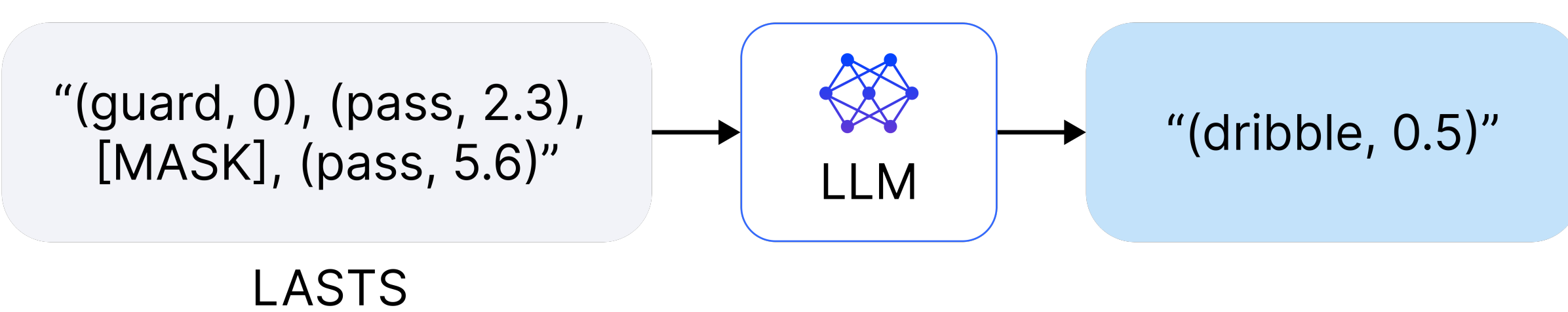
**Forecasting:** Predict the next event given a sequence of events



**Anomaly detection:** Predict the correct event type given a sequence of events with an incorrect event type

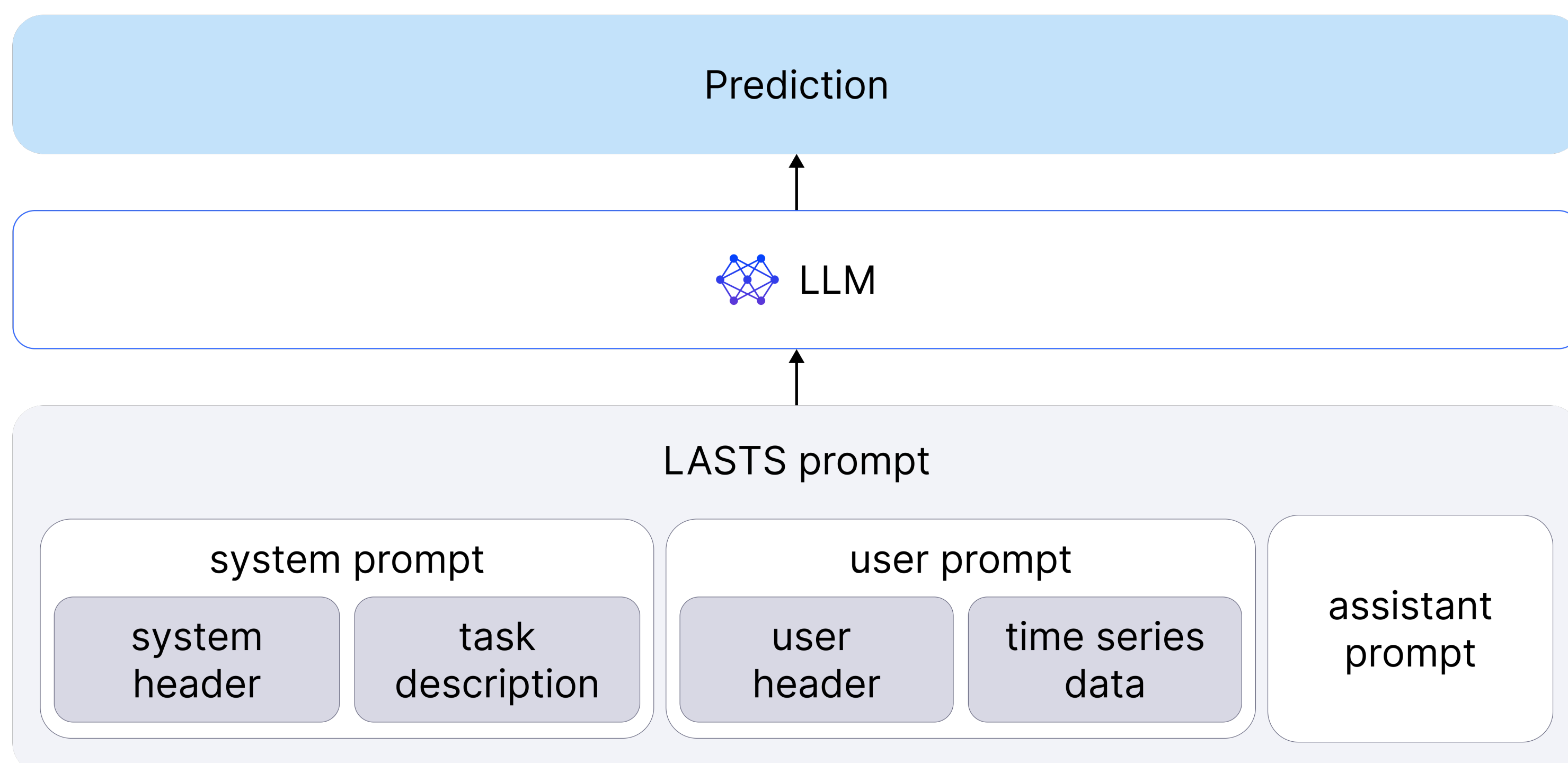


**Imputation:** Predict the masked event given a sequence of events containing a masked event



## Language-modeled Asynchronous Time Series (LASTS)

- Adapt LLMs to variable length asynchronous time series using natural language prompts
- Preserve LLM backbone with a prompting-based framework
- Flexible representation of events to handle many event types without predefined categories
- Support multiple tasks by defining them in the prompt
- Leverage LLM knowledge to enrich event semantics and strengthen temporal reasoning

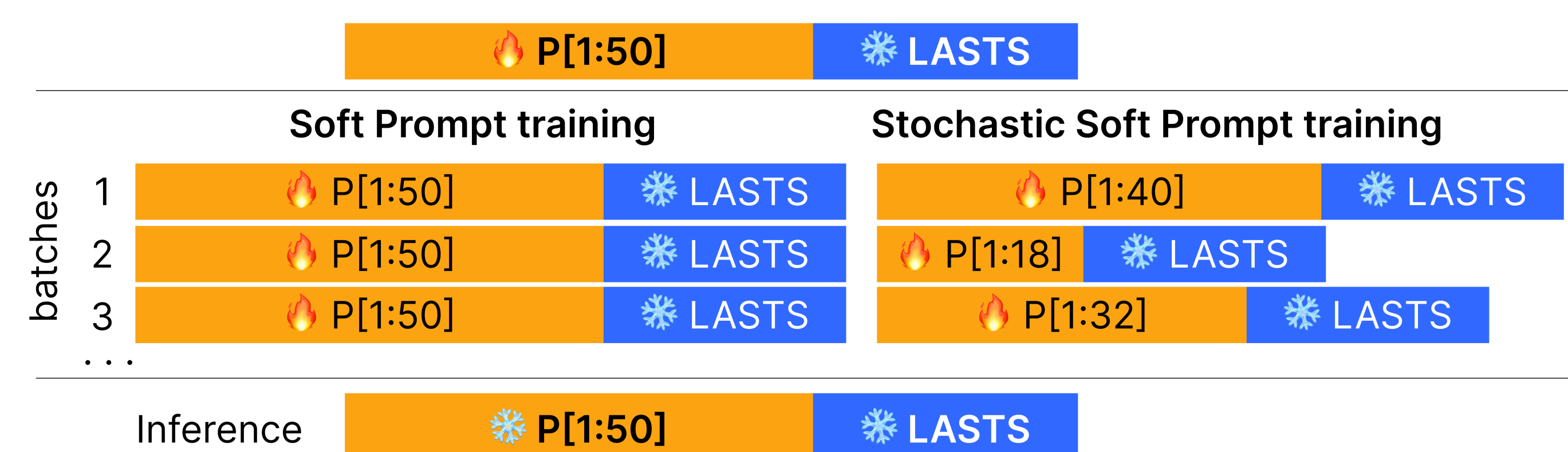


- System prompt:** description of the task and dataset
- User prompt:** asynchronous time series input as a comma-separated sequence of tuples
- Assistant prompt:** output generated by LLM

System prompt used for the forecasting task

You are a helpful assistant. Your task is to complete an asynchronous time series. [dataset\\_description](#). Each series is given in the format (inter\_arrival\_time, action\_name). This indicates that the action name started inter\_arrival\_times milliseconds after the start of the previous action or the beginning of time if it's the first action. The allowable actions are: [valid\\_vocab](#). Given the first few elements of an asynchronous time series, your task is to provide the next action with its inter arrival time as (inter\_arrival\_time, action\_name). You generate all your response as a single python tuple. Be sure to provide only that one python tuple and nothing else.

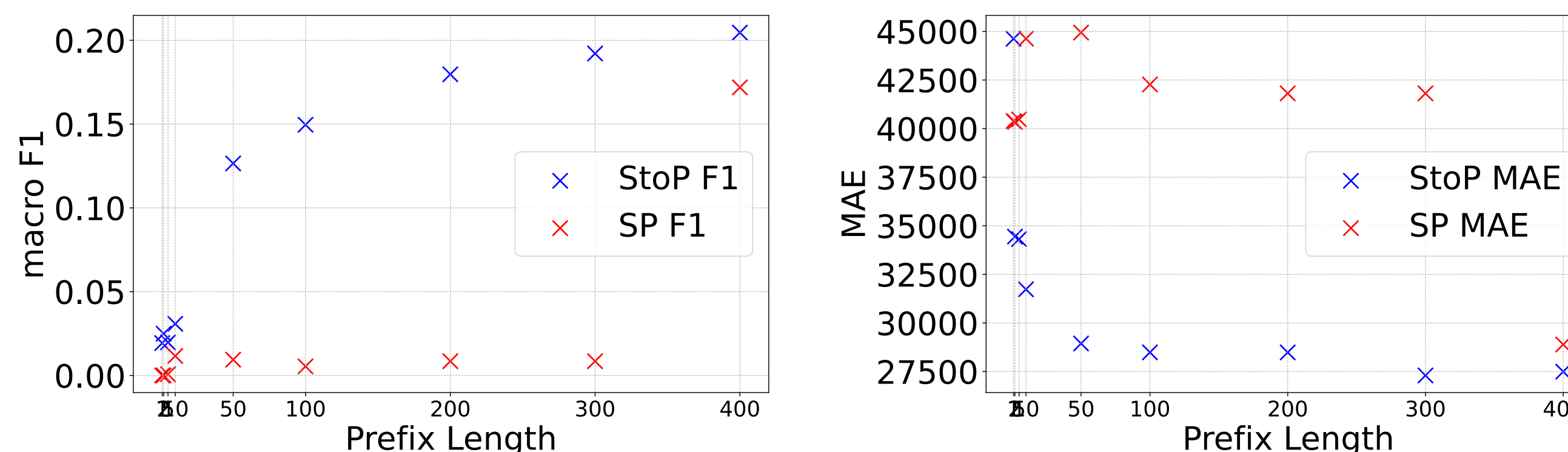
## Stochastic Soft Prompts (SToP)



- A novel prompt-tuning mechanism for parameter-efficient LLM adaptation
- Boost model performance and generalizability by randomly truncating prompts during training to encourage diverse representations

## Analysis & Interpretability

- All prefixes are valid prompts in SToP



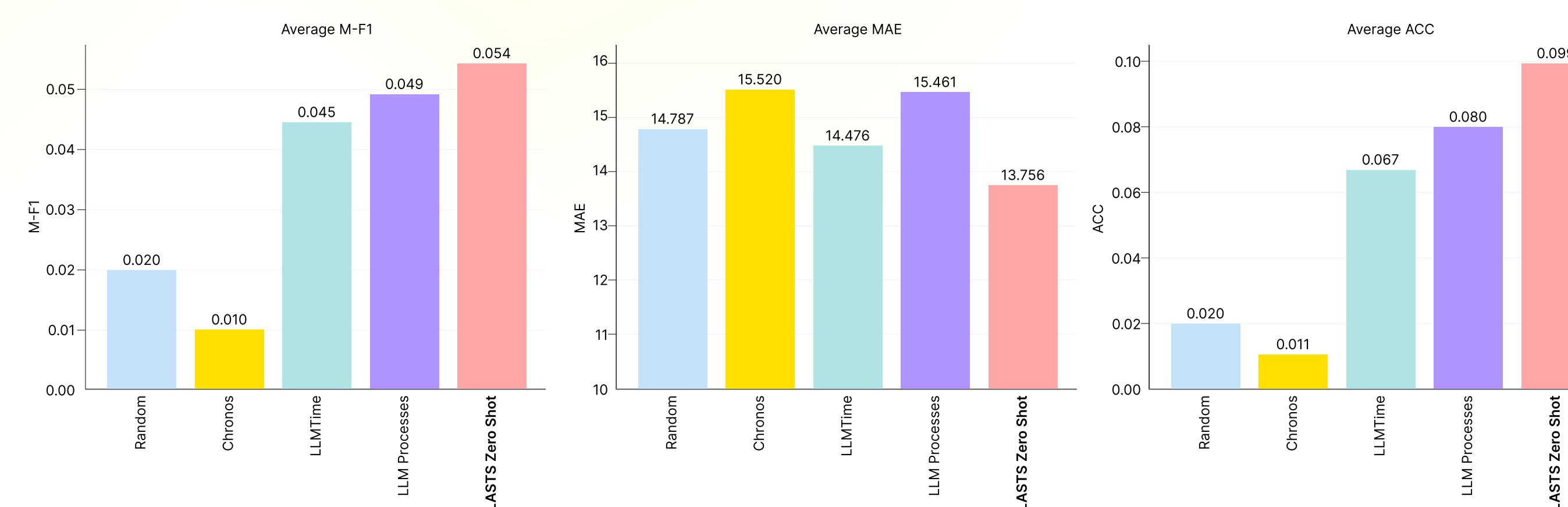
- SToP learned representations can be interpreted

Prompt: Tell me in as much detail as possible what task you are supposed to do

Output: The task appears to be related to preparing a snack or meal by using the facilities provided in the kitchen. Specifically, I was asked to perform a series of actions to accomplish this goal. Here's what I've observed:1. **Opening the cabinet**: I need to open the cabinet to access the necessary utensils, cookware, or ingredients.2. **Grabbing the ingredients**: I will need to pick up the specified ingredients, such as a bowl, a cutting board, or a can opener, to prepare the desired dish.3. **Slicing the ingredients**: I will cut the ingredients into the required proportions, shapes, or sizes depending on the recipe being followed.4. **Preheating the cooking surface**: If needed, I will turn on the stove, oven, or other cooking device to the desired temperature to start cooking the ingredients.5. **Cooking the ingredients**: I will proceed to cook the ingredients according to the required cooking method, such as boiling, sautéing, frying, or baking.6. **Adding condiments or seasonings**: If necessary, I will add the required condiments, spices, or seasonings to the cooked ingredients to enhance their flavor.7. **Combining the ingredients**: Finally, I will combine the cooked ingredients in the desired manner, whether it's mixing, blending, or assembling the final dish. Please let me know if I'm correct in my understanding or if there's anything else I can do to help!

## Results

- Llama-3-8B-Instruct; prompt length=400 (0.02% params)



Forecasting						
Model	Breakfast		MultiTHUMOS		EPIC-KITCHENS	
	M-F1 ↑	MAE ↓	M-F1 ↑	MAE ↓	M-F1 ↑	MAE ↓
Random	0.0162	40.1513	0.0417	1.8803	0.0000	3.2001
Chronos	0.0011	43.0502	0.0265	1.9805	0.0000	3.5925
LLMTime	0.0240	37.3902	0.1280	2.2060	0.0040	4.8948
LLMProcesses	0.0337	44.9856	0.1278	2.0471	0.0049	4.3843
LASTS Zero Shot	0.0604	38.1630	0.1361	1.8868	0.0105	3.1566
LASTS Few Shot	0.1518	35.5605	0.1676	1.8114	0.0149	3.3092
LASTS + QLORA	<u>0.2558</u>	33.9737	0.3218	1.7281	0.0764	<u>2.8964</u>
LASTS + SP	0.2341	<u>32.8417</u>	<u>0.3707</u>	<u>1.6630</u>	<u>0.0780</u>	<b>2.8830</b>
LASTS + StoP	<b>0.2633</b>	<b>32.5464</b>	<b>0.3947</b>	<b>1.6503</b>	<b>0.0797</b>	3.0318

Imputation						
Model	Breakfast		MultiTHUMOS		EPIC-KITCHENS	
	M-F1 ↑	MAE ↓	M-F1 ↑	MAE ↓	M-F1 ↑	MAE ↓
Random	0.0168	37.7029	0.0435	2.3622	0.0000	3.4269
Chronos	0.0013	38.4039	0.0294	2.3971	0.0000	3.6955
LLMTime	0.0137	35.9899	0.0968	2.6998	0.0005	3.6750
LLMProcesses	0.0156	34.7117	0.1123	2.3786	0.0008	4.2600
LASTS Zero Shot	0.0263	33.0097	0.0915	2.6696	0.0015	3.6527
LASTS Few Shot	0.0520	33.3440	0.1013	2.3982	0.0023	3.2528
LASTS + QLORA	0.1688	<u>28.5638</u>	<u>0.2132</u>	<b>2.2179</b>	0.0378	<u>3.1194</u>
LASTS + SP	<u>0.1581</u>	28.8503	0.2044	2.4092	<u>0.0423</u>	3.1456
LASTS + StoP	<b>0.2064</b>	<b>28.2251</b>	<b>0.2213</b>	<u>2.3445</u>	<b>0.0610</b>	<b>3.1116</b>

Anomaly Detection						
Model	Breakfast		MultiTHUMOS		EPIC-KITCHENS	
	M-F1 ↑	MAE ↓	M-F1 ↑	MAE ↓	M-F1 ↑	MAE ↓
Random	0.0349	—	0.0381	—	0.0238	—
LLMTime	0.0240	—	0.0415	—	0.0048	—
LASTS Zero Shot	0.0923	—	0.2755	—	0.0159	—
LASTS Few Shot	0.0837	—	0.3535	—	0.0337	—
LASTS + QLORA	<u>0.7011</u>	—	<u>0.6003</u>	—	<u>0.6520</u>	—
LASTS + SP	0.6520	—	0.5231	—	0.6159	—
LASTS + StoP	<b>0.7198</b>	—	<b>0.6045</b>	—	<b>0.6603</b>	—