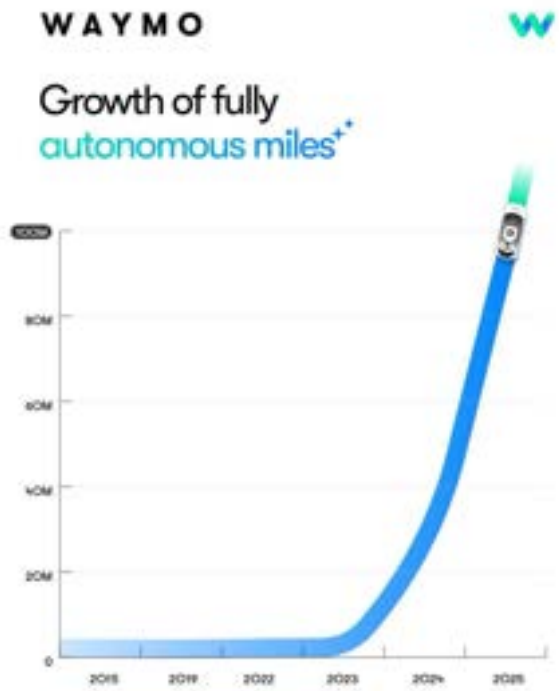
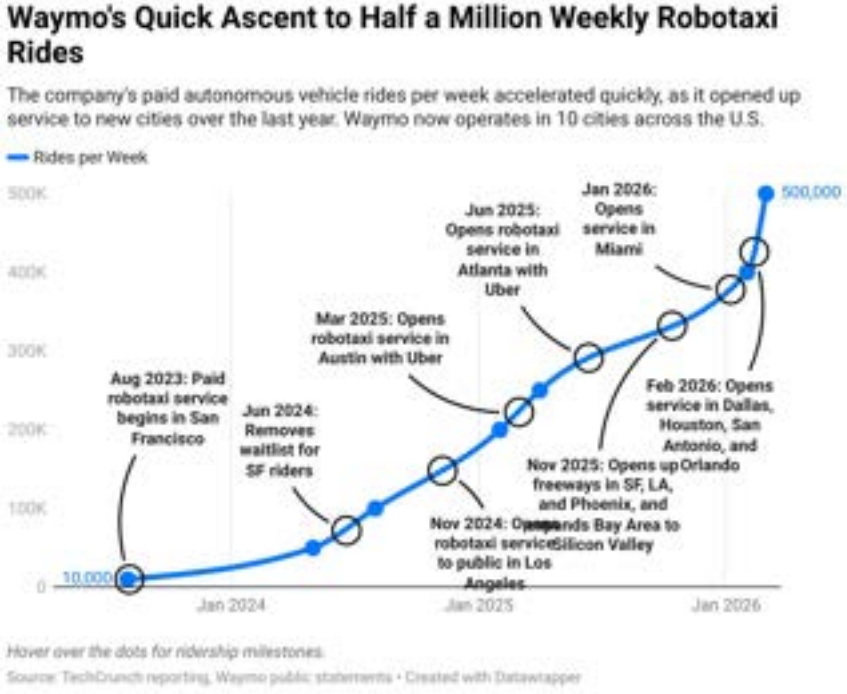
Time-Series Processing at Waymo

Joey Raso
Software Engineer @ Waymo

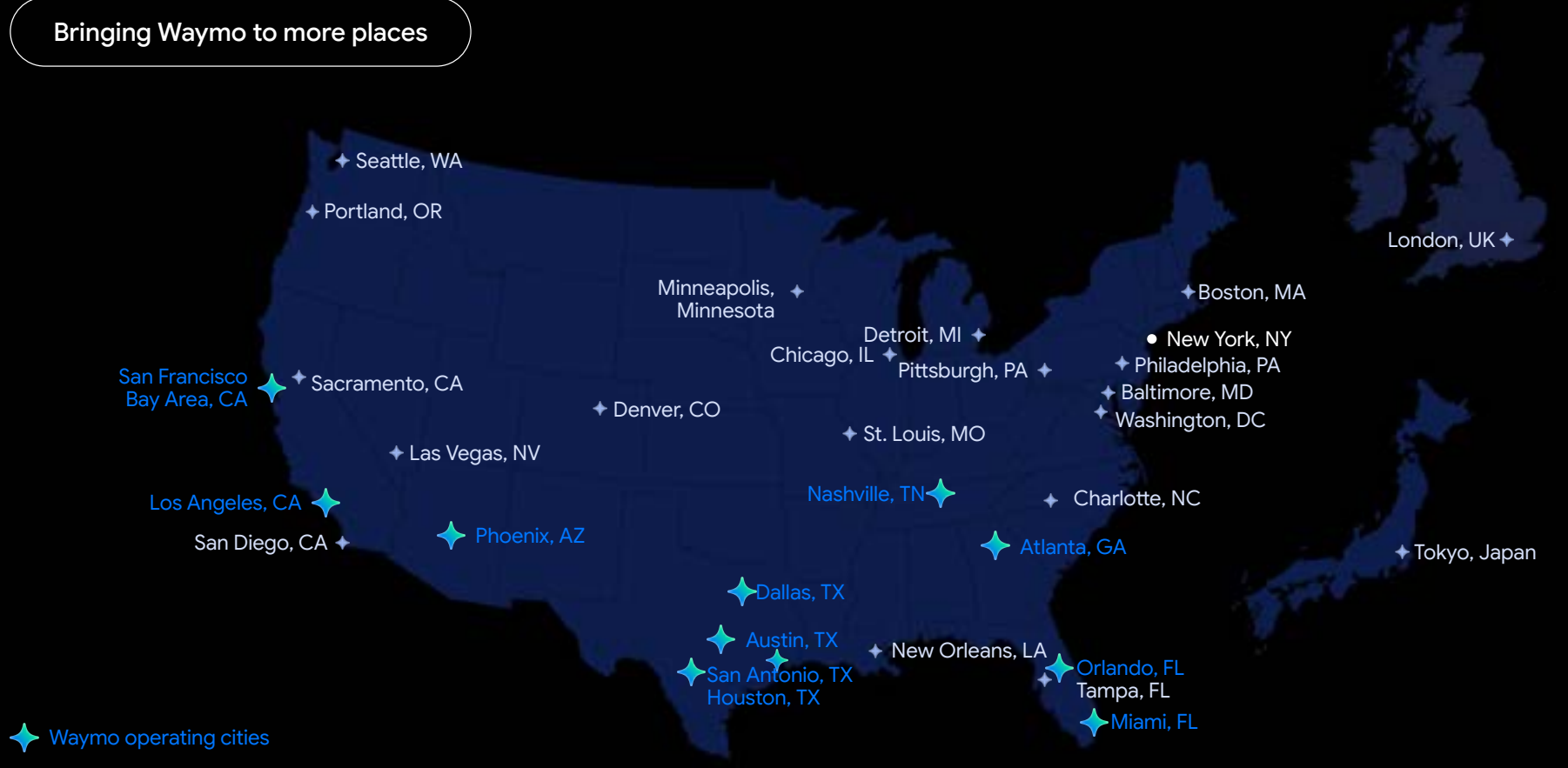
Waymo's Rapid Scaling





WAYMO

Bringing Waymo to more places

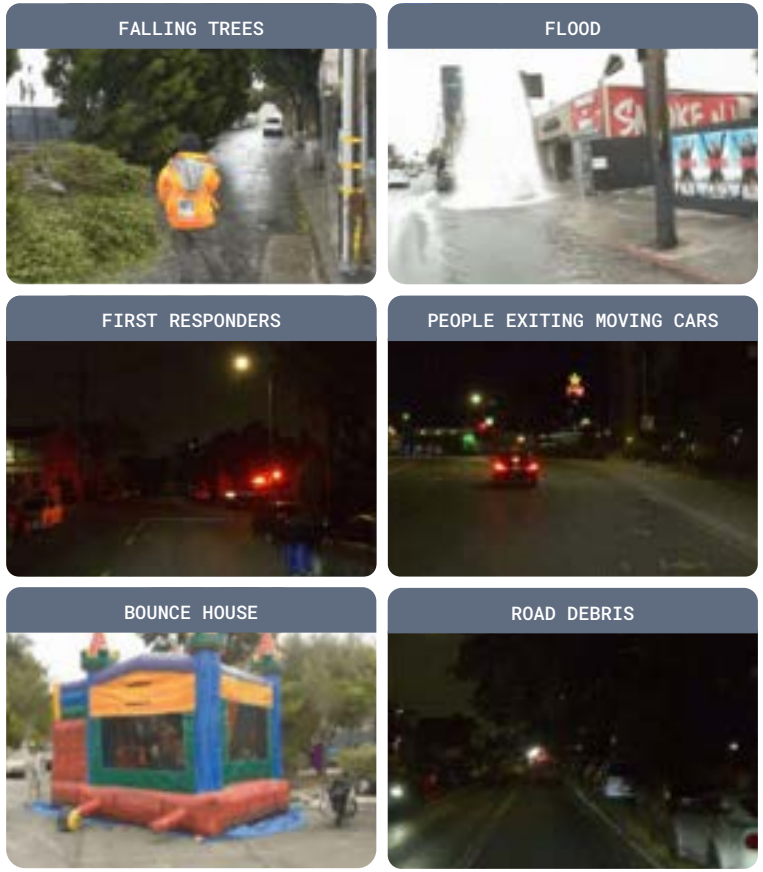


WAYMO

Early Event Discovery

As Waymo scales, we move further and further into the long-tail in terms of unique road conditions. This means that it's more important than ever that we're collecting interesting road events for further analysis.

Further, we'd like this collection to happen as quickly as possible to establish confidence as we roll out new software to the fleet.

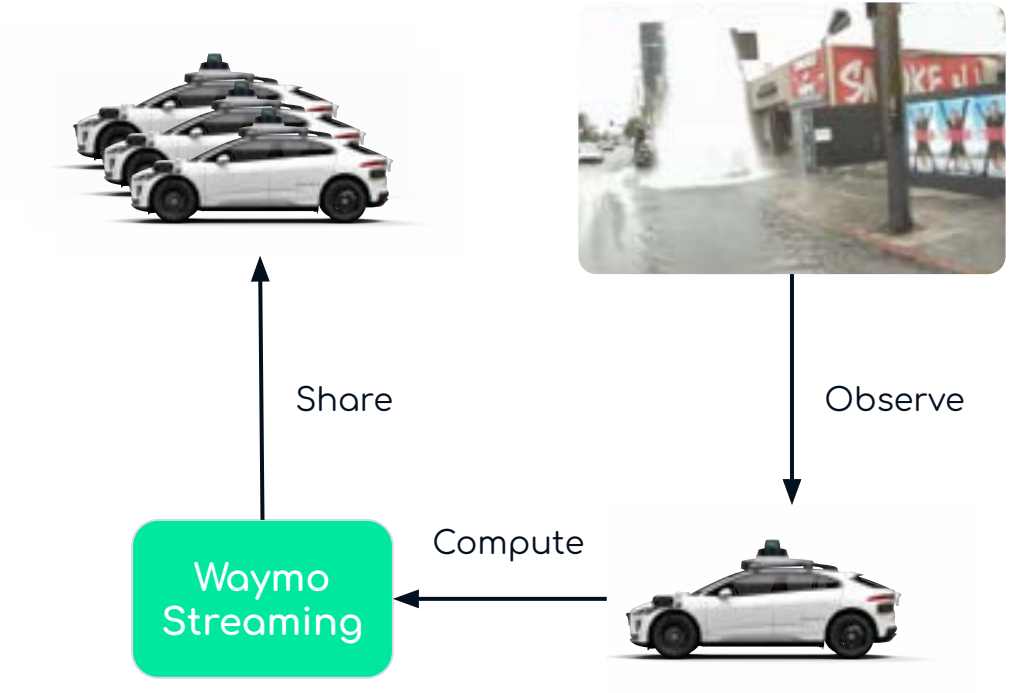


WAYMO

Fleet Information Sharing

A higher concentration of vehicles on the road means that there are likely to be road features that we are travserving multiple times per day across the fleet.

By propagating and sharing information amongst the fleet, we can approach tricky situations with a stronger prior and enhance the Waymo Driver's ability to respond.

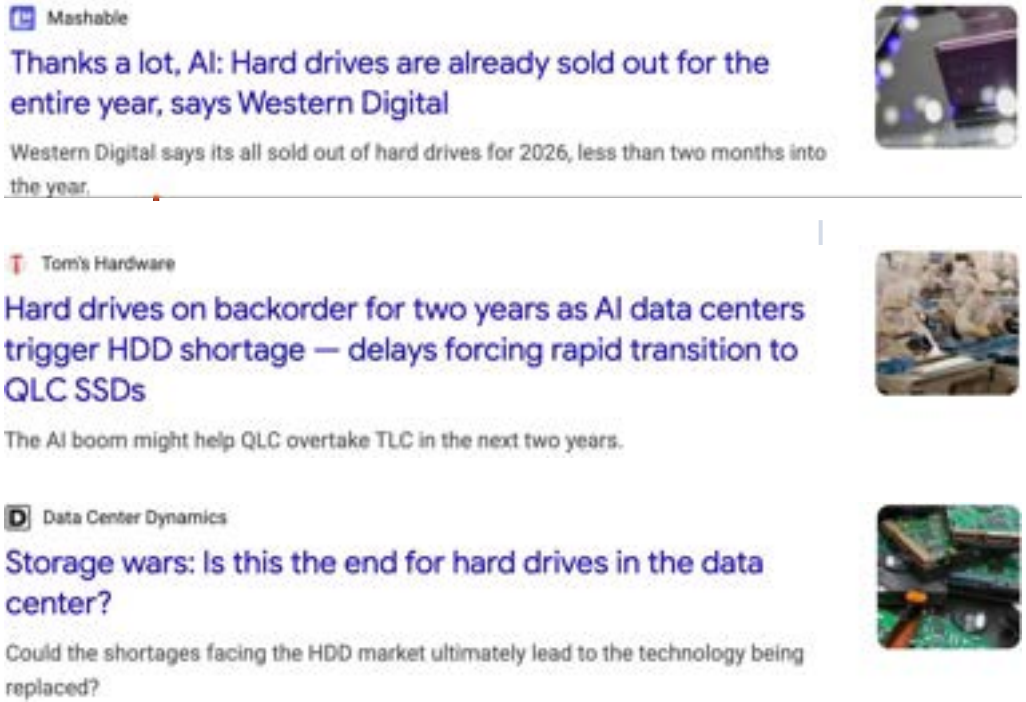


WAYMO

Importance Sampling

While Waymo has access to a massive amount of data storage, as we've solved the challenges associated with autonomous driving, we try to be efficient with our data storage usage and are seeing opportunities for further efficiencies as we scale.

As we scale and drive millions more miles in more places, we need to build systems to refine and grow our datasets to reflect the world we're driving today and tackle the remaining long-tail challenges while ensuring an efficient approach to data storage.



WAYMO

To enable these Early Detection, Observability, and Importance Sampling use cases, we need a capable Streaming Data Processing Framework to **process the Waymo driving data**, **make that data available to our engineers**, and, when applicable, **propagate signal to the fleet.**

WAYMO

System Requirements

Fleet Scale: Our Streaming systems should be able to handle the data load of Waymo's rapidly expanding fleet.

Late-Arriving / Out-Of-Order Data: The fleet data is subject to network latencies and other delays. Whatever system we build needs to be able to handle these delays gracefully.

Low Latency: To get the most out of our data, we need to get it in front of our engineers as quickly as possible. Therefore, the system must be able to process at low latency.

Stateful Processing: Much of the signal that we need to generate relies on timelines / windows of data rather than point observations. The framework should have primitives for stateful processing.

Hypothetical: Red Light Runners

WAYMO

Hypothetical: Red Light Runners

```
perceptions = (
    p
    | "ReadStream" >> beam.io.ReadFromPubSub(
                subscription=input_subscription)
)

red_light_runners = (
    perceptions
    | "WindowIntoFixed" >> beam.WindowInto(window.FixedWindows(5.0))
    | "KeyByObjectId" >> beam.Map(lambda obj: (obj.object_id, obj))
    | "GroupTrajectories" >> beam.GroupByKey()
    | "CalculateKinematics" >> beam.ParDo(DetectRedLightRunnersFn())
)

red_light_runners
    | "EmitSignals" >> beam.Map(print)
)
```

Consume Data Streams from Waymo Fleet

Aggregate and perform Kinematics Calculation

Emit to your Sink of choice

WAYMO

Hypothetical: Red Light Runners

With our new Dataflow-powered Red light runner detector, we could:

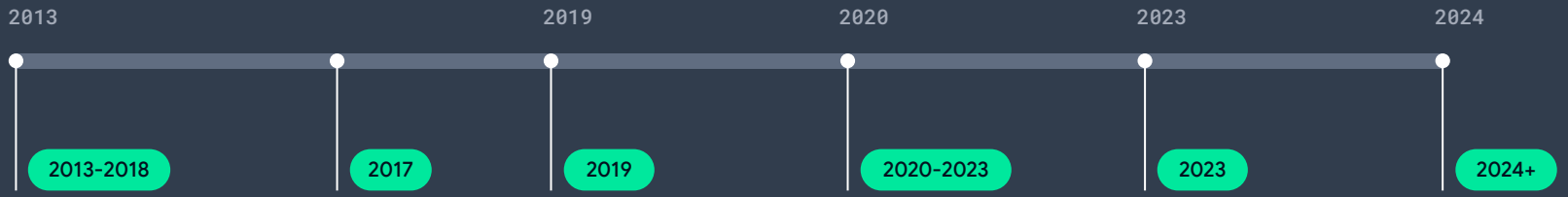
- **Discover red light runner events** in the logs to generate test sets.
- Mark log segments containing red light runners for **extended retention**.
- **Share these events event** back to the fleet to inform the Waymo Driver of potentially dangerous intersections.

LLM-powered Signal Generation

We've seen how we can produce low-latency signal to power our Fleet Information Sharing, Event Discovery, and Importance Sampling use cases. As is the case with out Red Light Runner detector, our signals have traditionally been heuristics-based. Recent industry-wide AI developments, combined with Waymo's AI expertise, have opened up the door for AI / LLM-Powered Signal Generation.

WAYMO

Waymo is a pioneer in AI for autonomous driving



Deep Convolutional Networks with primary use in Perception

Transformers advancing perception and behavior modeling

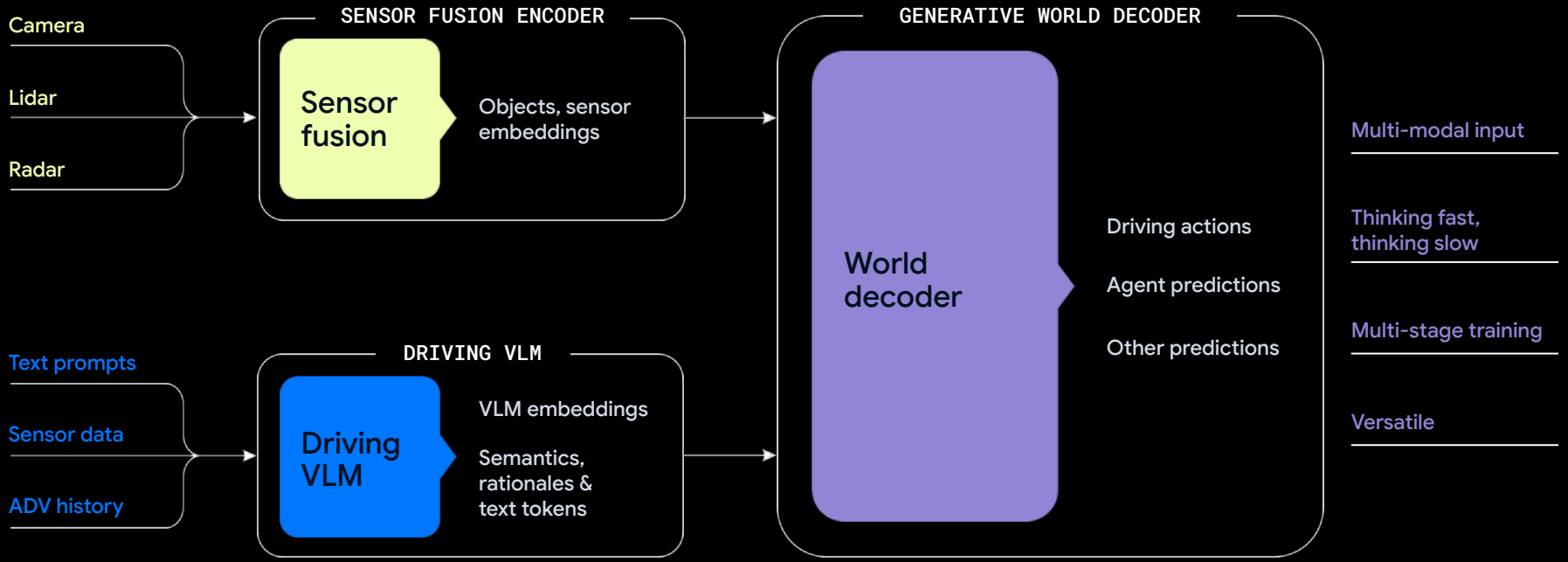
Waymo Open Dataset released, the most comprehensive dataset for autonomous driving research

SOTA Deep Learning widely used across Perception, Planning, and Simulation

VLM / LLMs boosting the performance of the Waymo Driver with general world knowledge

Waymo Foundation Model driving our next generation AI stack by combining AV domain knowledge and general world knowledge of VLM/LLMs

The Waymo Foundation Model



LLM-powered Signal Generation

Forward sensor + camera data over the cell network to Waymo's compute fleet.

The Waymo Streaming System performs windowed aggregations to generate Waymo FM features.

Waymo FM generates World Embeddings to as features in our importance classification.

WAYMO

Hypothetical: Red Light Runners

```
perceptions = (
    p
    | "ReadStream" >> beam.io.ReadFromPubSub(
                subscription=input_subscription)
)

red_light_runners = (
    perceptions
    | "WindowIntoFixed" >> beam.WindowInto(window.FixedWindows(5.0))
    | "KeyByObjectId" >> beam.Map(lambda obj: (obj.object_id, obj))
    | "GroupTrajectories" >> beam.GroupByKey()
    | "RedLightRunnerDetector" >> beam.ParDo(RunnerDetector())
)

red_light_runners
    | "EmitSignals" >> beam.Map(print)
)
```

Consume Data Streams from Waymo Fleet

Aggregate and perform Inference

Emit to your Sink of choice

WAYMO

Hypothetical: LLM Red Light Runners

- Composable APIs allow us to drop in an LLM Inference with minimal disruption to the rest of the pipeline.
- ML-first APIs enhance our heuristics-based approach with LLM Classification. With this API we can easily hook into Waymo's fleet of GPUs and TPUs.
- A combination of Batch and Streaming Execution modes allow us to easily backtest our model improvements.

WAYMO

QUESTIONS?

WAYMO