



# Autonomy at Scale: Multi-Policy Decision Making (MPDM) v. Early Commitment in Autonomous Driving

JACOB CROSSMAN / MARCH 20, 2023  
Vice President of Autonomy



## Table of Contents

- 3 Traditional Approaches to Autonomous Driving
- 4 A Deep Dive Into Early Commitment Decision-Making
- 6 The Problem: Decision-Making Under Uncertainty  
The Solution: Multi-Policy Decision Making (MPDM)
- 9 MPDM Overcomes Uncertainty and Enables Scaling

## Traditional Approaches to Autonomous Driving

Most approaches to autonomous driving use some form of **early commitment** to make their decisions. Early commitment means that the logic used to make driving decisions is **fixed** and **determined in advance** of actually encountering a scenario.

Early commitment is *reactive*. Driving is frequently reactive, so this type of decision-making often works, especially in common situations. For example, drivers use early commitment to respond to a red light ahead. We know from driver's training and experience that we should stop in front of a red light and not proceed through until it changes to green. We don't need to think about it; our reaction to the light is "hardwired." We can even be doing another task, such as talking with the passenger next to us, and still respond correctly.

The simplicity and applicability of this approach to driving makes early commitment the first and often the only approach engineers take to making a car drive itself. Early commitment is typically implemented in two ways:

1. Engineers directly encode decisions in a form similar to "if/then" rules (e.g. "if there is a red light ahead, then stop by some predefined point").
2. Engineers train the system to respond to the same situations using machine learning. In this case, the learning process generates the equivalent of the rules that a programmer would create by "observing" (being trained on) a wide range of scenarios. For example, the system could be trained on dozens or hundreds of approaches to red lights at different speeds and distances.

Though they appear different on the surface, both of these approaches hard code how the vehicle will respond to every situation it encounters before ever driving its first mile. These approaches share the advantage that they can respond rapidly and correctly to many situations. However, they have two important disadvantages that limit their power in practice.

1. They **cannot reliably handle situations that have not been anticipated** by the engineers that built them and that do not exist in their training sets.
2. They are **resource intensive to build and maintain**, making them difficult to scale to different environments and situations.

To get a better understanding of these limitations and how May Mobility overcomes them, we dive a little bit deeper.



## A Deep Dive Into Early Commitment Decision-Making

Consider the scenario presented in Figure 1 at right. The ego vehicle (in green) is approaching an intersection with two vehicles (in blue) and a pedestrian (teal circle).

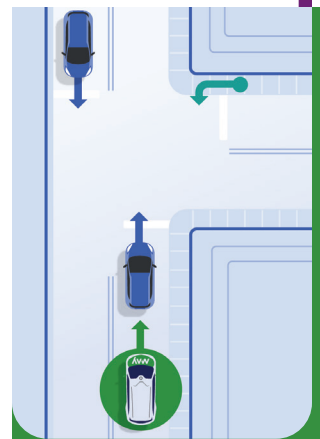
In an **early commitment system**, the engineer creates an algorithm that selects how to respond based on the situation it senses. Two examples (a real system would have many more) of how this could be done include:

1. Selecting the nearest forward object to follow (i.e. the car in front of the ego)
2. Selecting the object near the route that poses the highest risk (i.e. this might be the pedestrian, depending on where the pedestrian is standing with respect to the road)

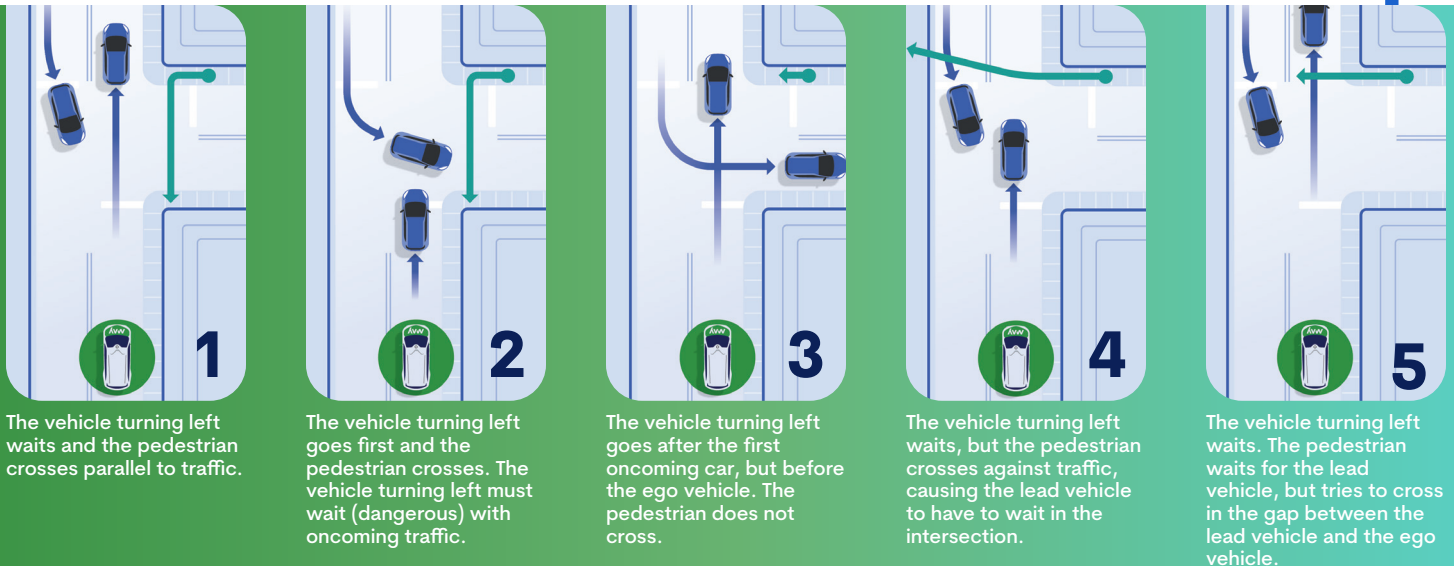
Instead of an engineer directly programming these decisions, the system is presented with many scenarios (e.g. in simulation) and rewarded or penalized depending on whether the outcome was safe ("good") or dangerous ("bad").

The problem with this approach is that there are many ways this scenario could play out. In Figure 2 (below), we show five distinct possibilities, from relatively benign cases to some tricky stopping scenarios, though these are by no means exhaustive.

When programmers determine what the system should do before it ever encounters a situation, there is the potential that this decision will be wrong and lead to bad results. For example, what if scenario 5 occurred while the system was following the lead vehicle? Or what if scenario 2 happened while the vehicle was reacting to the pedestrian? **What if a scenario the engineers never considered occurs?** This last situation is, in fact, likely, as the combination of things that can occur in the real world is extensive.



**Figure 1:** A driving scenario. The autonomous vehicle (ego) is green. Two other vehicles are shown in blue. The teal circle represents a pedestrian. The arrows show the recent movement trajectories for each object.



**Figure 2:** The scenario from Figure 1 can play out in many ways, including these 5. It is difficult to write logic that anticipates every possible scenario. It is also difficult to predict which one will occur and thus select the correct action.



To help alleviate these issues, engineers frequently use prediction to help increase the probability that the action selected best fits the situation. Instead of selecting an action based on where each object is and how it has moved in the past, the system selects actions based on where it computes that objects will be in the future.

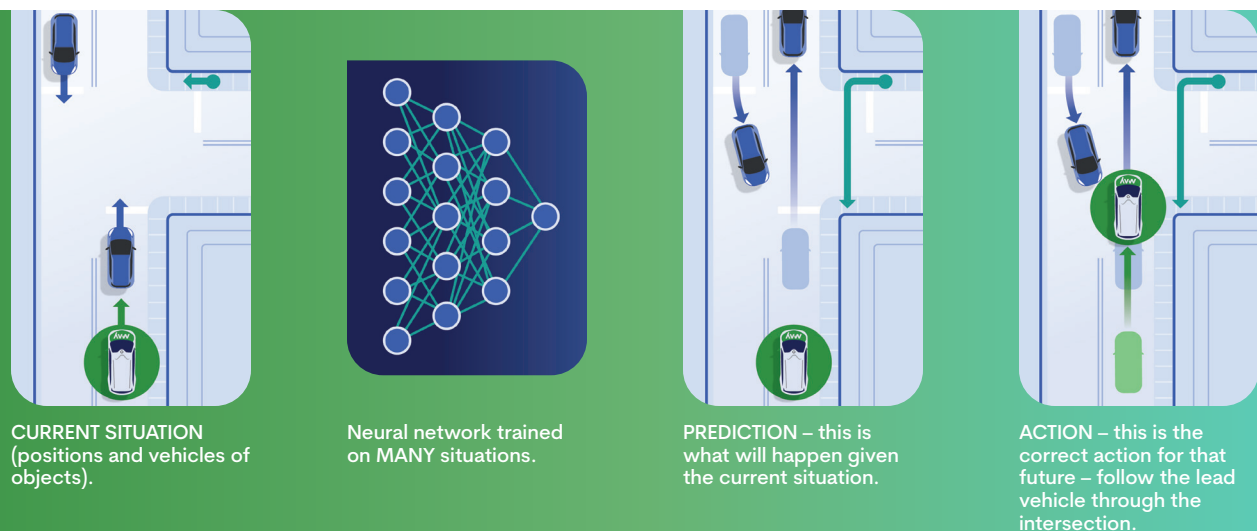
For example, the system might process the positions and velocities of every object in the scene using a neural network that makes a prediction about how the scenario will play out. In this example, the system might find that Scenario 1 from Figure 2 is the most likely, based on the cases observed during the network training.

Typically, this type of approach gives better results, but it is still fundamentally an early commitment approach. The selected action is based on a prediction instead of the current observed state, but this action is still pre-determined by the engineer or machine learning algorithm.

For example, a predictive system will stop for the pedestrian if it predicts the pedestrian will enter the crosswalk in the future rather than waiting to see it move into the crosswalk. However, the stopping action is still hard coded and the system is still prone to make mistakes when it encounters complex interactions between different entities (e.g. Scenario 4 in Figure 2 is particularly difficult for these systems) or because the scenario is not observed in past training and test data.

To solve these problems, autonomy teams must expand their engineering teams and exponentially grow their data needs. The thought is that if enough data is collected, eventually the system will see every situation and will respond correctly to them all.

In practice, the space for possible action combinations is vast, and even small environmental changes such as road widths, vegetation positions, parking spot positioning, crosswalk locations, etc. require significant engineering effort and data processing to handle correctly. **Fundamentally, systems based on early commitment do not scale well.** They are resource intensive to build and maintain.



**Figure 3:** Some autonomous systems use machine learning to predict what will happen. This can greatly improve performance; however, when the system is wrong, it can result in bad or dangerous behavior.



## The Problem: Decision-Making Under Uncertainty

The challenge with early commitment systems is that they depend on **knowing with certainty** what will happen. These systems must know this information in order to select the correct action. This requirement for certainty is why so much time and effort is spent perfecting expensive perception systems, gathering petabytes of training data, and processing millions of scenarios. Uncertainty must be driven to near zero.

What if we didn't need to be so certain? What if there was a way to make safe and effective decisions, even when there is uncertainty? Such a system would be much simpler and easier to scale. It could also successfully navigate scenarios that it had never before observed. It turns out that there is a way to do this; humans do it all of the time.

Humans use **deliberate reasoning** to navigate scenarios like those shown in Figure 1. When a human encounters such a scene, the driver slows their secondary actions (i.e. talking with a passenger), slows the vehicle, and begins focusing on predicting what will happen. Human drivers will rapidly run "what ifs" through their minds. What if the pedestrian moves in front of the lead vehicle? What if that oncoming car turns first? What if the oncoming car turns in front of my car? While the human driver remains uncertain, their actions remain cautious and speeds remain slow. After each other person in the scene commits to their action (whether to turn, whether to cross, or whether to go), a human driver will become more confident and act with more assertiveness.

Can a computer system be designed to behave similarly in uncertain situations? The answer is yes, and doing so unlocks the potential for behavior that requires far fewer resources to build while responding safely to a much larger set of situations.

**Multi-Policy Decision Making (MPDM)** is designed specifically to solve the challenge of making safe driving decisions under uncertainty, every time, even when having never encountered a scenario before.

## The Solution: Multi-Policy Decision Making (MPDM)

MPDM avoids issues described previously by making decisions that account for **all plausible scenarios** without requiring early (premature) commitment to any specific actions. This approach is called a **least commitment** approach because the system leaves its options open as long as possible and makes a final decision about which action to take at the time it needs to act.

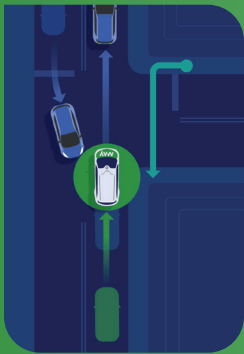
In an MPDM system, actions are **not hard coded** in advance to fit a specific set of scenarios. While the system does make future predictions, **it does not make the assumption that it knows which future will occur**. Instead, MPDM imagines many futures and picks actions that are safe no matter which future occurs.



To show how MPDM accomplishes safer driving in uncertain situations, consider Figure 4. MPDM begins with what the perception system provides – the positions and velocities (over the past several seconds) for every *agent* (e.g. cars, people, cyclists, etc.) in the scene. For each of these agents, MPDM generates a set of plausible actions for it. Here “plausible” means that it is physically possible given the environmental constraints and consistent with the movement patterns observed in the past few seconds. These actions are not always *likely*, which is important because sometimes the high-risk actions are not probable, but must still be considered.

MPDM then selects a set of potentially useful actions (called **policies**) for the ego vehicle (in green). None of these actions are selected specifically to fit the scenario, they are **general behaviors** that might be useful in a wide range of situations. For example, actions include “go the speed limit while following a lead vehicle,” “brake for the pedestrian in front,” “brake before the intersection,” “go a little slower,” “shift a little to the left,” etc.

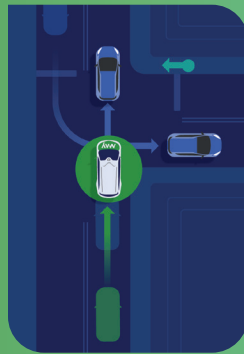
#### Policy Option 1: Drive the speed limit or match the lead vehicle speed if it is driving slower.



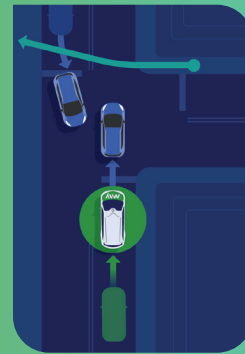
No issue if this occurs.



A hard stop if this occurs.



A close call, but no collision if this occurs.

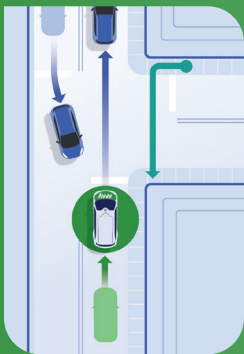


A medium stop if this occurs.

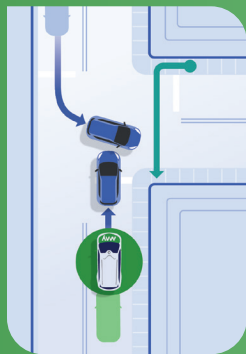


Accelerate at a pedestrian (close call) if this occurs.

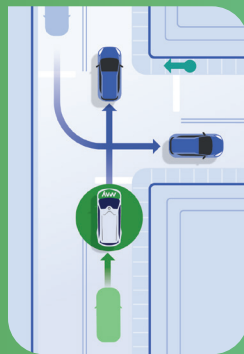
#### Policy Option 2 (Best Option): Drive a little bit slower (e.g. 3 mph) than the lead vehicle, while still reacting if it stops.



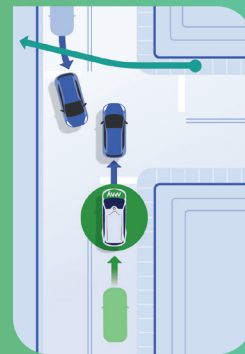
No issue if this occurs.



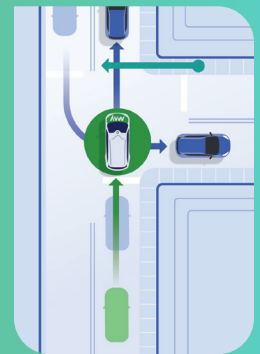
A medium stop if this occurs.



A comfortable gap for the left turn vehicle if this occurs.



A normal stop if this occurs.



Pedestrian crosses before vehicle passes if this occurs.

**Figure 4:** An example of how MPDM imagines many futures for each policy that it can execute. Policies are algorithms that control vehicle behavior in particular ways. (e.g. slowing, accelerating, veering, stopping). Here, for simplicity, we show just two policy options (Option 1: driving the speed limit and Option 2: driving a little slower), but in practice MPDM considers 6–12 policies every times it makes a decision. For each of these options many futures are imagined (i.e. **simulated much faster than real time**). Here we show 5 scenarios that are simulated, but in practice hundreds of options are simulated.



## THE NEXT TWO STEPS ARE THE CRITICAL PIECES THAT SOLVE THE DECISION-MAKING PROBLEM.

### STEP 1:

MPDM selects one action for each object in the world and one policy for the ego vehicle. It then **simulates the future** that would occur if these objects all behaved using the selected actions. However, MPDM doesn't stop here. The system cannot be certain that the future it simulated will really happen. What if different actions are taken by the agents? So, it picks another, different set of actions and simulates those. In fact, it does this many times, hundreds of times. In the end, MPDM has simulated many, many futures and created a range of outcomes, none of which are certain, but all of which are plausible. Effectively, the vehicle is **virtually driving through that specific scene hundreds of times**, sometimes making mistakes, sometimes succeeding and learning the best way to get through it in real time.

### STEP 2:

MPDM **evaluates all of the imagined futures** for each policy the ego vehicle could execute. If any policy generates a plausible future where something bad happens (e.g. a collision or close call), it is rejected. For any remaining policies that are not rejected, the one that provides the most comfortable progress toward the driving destination is executed.

Steps 1 and 2 are repeated five times per second. Thus, **as the world changes** (e.g. a new obstacle is detected or a vehicle changes its direction or speed), **it immediately responds** with the best policy for that updated situation.

APPROACH	HOW DECISIONS ARE MADE
Early Commitment	<b>Rule-like (programmed or learned)</b> If observe scenario A, then do behavior Z.
Early Commitment with Prediction	<b>Prediction with Rule-like algorithms</b> Predict future state of all objects using a learned model.  If <b>future</b> looks like scenario A, then do behavior Z.
HOW MPDM COMPARES	
Regardless of the observed scenario:  Use a fast simulation to predict all plausible future states of all objects when the ego vehicle does policies X, Y, and Z.  Execute whichever policy is both safe and most comfortable.	

**Table 1:** How MPDM compares against more traditional early commitment approaches.



# MPDM Overcomes Uncertainty and Enables Scaling

On the surface, MPDM appears simple and, in some ways, it is. To make MPDM work we must solve two hard problems: creating models of human behavior in driving situations and creating a simulator that can run hundreds of times faster than real time. However, once these problems have been solved its simplicity belies a power that overcomes the challenges that novelty, uncertainty, and scale present to early commitment systems.

First, MPDM can handle any situation for which it can model human behavior, **even novel situations that it has never previously experienced**. The reason MPDM can handle novel situations well is that its decisions do not depend on identifying the situation. MPDM always considers **every action** that might apply and virtually drives that scenario before selecting it. The best action emerges from the virtual drives.

Second, **MPDM chooses safe actions even when there is uncertainty in the environment**. If, because of occlusion, the perception system cannot determine if a person is a pedestrian or a cyclist, MPDM simulates both possibilities. If the tracking system cannot determine perfectly which of two lanes a vehicle is in, it simulates its possible actions in both cases. If the system cannot determine with certainty whether a pedestrian will cross the street or loiter at the curb, MPDM makes sure its selected action is safe in either case. As a result, the behavior of an MPDM system is much like a human – it slows and gives more room when it is uncertain and drives more assertively when it is more certain.

Finally, **MPDM scales much better and requires many fewer developers and resources to build** than early commitment approaches. This scaling advantage derives from the fact that it is not necessary to recognize or model combinations of things when programming MPDM. Combinations of things (e.g. the light turning yellow while following a car with a pedestrian possibly crossing in front) results in an explosion of possible situations which, if modeled directly, require significant engineering and data processing resources. With MPDM we are only required to create models of human behavior in driving situations, which is a simpler problem. The construction of this model can take significant effort and time, but once it is functional and verified, it can be applied to all environments and situations. MPDM is the most scalable approach to developing autonomous behavior.

Taken together, these three advantages **provide the building blocks for scale**. If a technology can be built efficiently to work well in novel and high-complexity situations and limit site customization, it can rapidly be deployed to thousands of different environments. MPDM is a unique technology that provides these advantages and unlocks autonomy at scale.

It is the future of autonomous driving.

