Achieving Safe Autonomous Vehicles: Technical and Non-Technical Solutions

Michael S. Emanuel
Dylan Randle

Motivation

Autonomous vehicles (AVs) are a high stakes growth area for machine learning. Safe driving is quite literally a matter of life or death, with car accidents estimated to kill approximately 37,000 Americans and a staggering 1.25 million people annually around the world. [1] Money is also pouring in, with an estimated $4.2 billion invested in the first three quarters of 2018.[2] Intel has estimated that the autonomous vehicle industry will create $800 billion of annual revenue by 2035 and $7 trillion by 2050, and that the technology will save 585,000 lives between 2035 and 2045. [3]

AVs hold the promise of reducing traffic accidents while improving transportation efficiency. However, their widespread implementation has faced significant difficulties. There seems to be very little, if any, tolerance for failures. Even while human drivers in the US killed over a hundred people a day in driving accidents in 2017, machines are being held to a higher standard.

Some industry leaders have argued that this is an irrational double standard and that autonomous systems should be rolled out rapidly as soon as their safety performance is on a par with human drivers.[4] The lives saved by this technology, the argument goes, will more than offset the lives lost in accidents caused by AVs as they improve. While this argument may be correct from an arithmetic standpoint, we disagree with that approach and believe AVs will face an intense backlash unless they are introduced slowly, carefully, and in a way that persuades drivers and pedestrians that the vehicles are meaningfully safer than human operated vehicles.

Drago Anguelov, Principal Scientist at Waymo, argues that one of the primary challenges in building safe AVs is gathering more training data that captures all the very rare events that can happen on the road, and then training machine learning models to handle these correctly. Anguelov titled his talk “taming the long tail” of autonomous driving. He believes that AV software needs to be able to predict what various actors (pedestrians, other drivers, other AVs) will do in *any* given driving situation the car encounters. At the same time, his team is restricting driving experiments to small, controlled environments and only very slowly expanding to other areas.

We believe that a combination of both technical and non-technical solutions will be the way forward. In this project we explore this problem and, leveraging techniques we have learned in

---

[1] NHTSA Fatal Motor Crashes 2017 WHO Road Traffic Deaths 2013
[2] Axios Autonomous Vehicle Investment 2018
[3] TheVerge - Intel Predicts $7 Trillion Autonomous Vehicle Industry
[4] Elon Musk has commented along these lines, before and after two fatalities in Tesla vehicles operating in “autopilot” mode

the course as well as exciting new areas of distributed and secure machine learning research, propose new solutions.

Background
Tesla has had great success at convincing loyal fans to drive about a billion miles in semi-autonomous mode, while Waymo has racked up about 10 million fully autonomous miles. But to put this scale in context, Americans drove approximately 3.2 *trillion* miles in 2017, or 8.8 billion miles per day. If large numbers of people could be persuaded to buy new cars that harvested their driving data to the train neural networks used in AVs, the effects could be transformational for the industry.

This is where privacy and protecting proprietary data come to the fore. How many people would allow a big company like Google, GM or Audi access to a data dump of all their driving? What happens if you get into an accident, or just run a red light, to say nothing of the privacy of your itinerary? Once the data is there, it can be subject to subpoena by the government.

Companies have also made significant investments in their technologies. Imagine a platform that allowed start-ups to train their driving algorithms on a fleet of vehicles equipped with sensors. This could open up competition in the image recognition and scene segmentation niches of the AV market. It could also allow for economies of scale, with competitors working on different models while sharing access to the same data platform, rather than cutting expensive exclusive deals to obtain data.

Technical Solutions
One technical solution we propose is to leverage privacy-preserving and secure computing frameworks to enable the unlocking of very large, private training data that will better cover the rare, yet important, driving scenarios that AVs must learn to navigate safely.

The challenge here is three-fold: ensuring the privacy of individuals' driving data is protected, distributing training to the various (possibly a very large number) of decentralized data sources, and protecting the proprietary information of the models (weights, architectures). We endeavor to implement a recently-proposed distributed training stack that combines federated learning (FL), differential privacy (DP), and secure multi-party computation (SMPC) to allow the training of proprietary models on very large, distributed driving datasets without the need to directly expose sensitive data. Our goal is to explain this new training regime, as well as to analyze its performance in terms of accuracy and execution time with respect to a baseline centralized, non-secure dataset.

Non-Technical Solutions
How can autonomous vehicles be deployed safely before they are perfect? How can companies in this industry act responsibly to maintain safety and the community's trust without slowing development too much? What is the appropriate regulatory framework for AV's? What is the optimal pace at which AV technology should be tested and deployed on public roads? We will address these questions below, using the analytical techniques we have developed in AC 221. We will consider these questions largely in the context of case studies of two of the leading companies developing fully autonomous vehicles: Waymo and Uber. These companies have taken quite different approaches in the past (though recently Uber has been shifting closer to Waymo's style), and many complex issues can be clarified by considering the relative merits of each company's strategy.

Training Neural Networks without Retaining Sensitive Data

Machine learning and neural networks have driven major advances in artificial intelligence in the recent past. They have led to performance breakthroughs in a range of tasks including image recognition, machine translation, and complex strategy games such as Go and Chess. But the term "deep learning" is often thrown around as a buzzword by people who don't know what it means, and many technically proficient people who haven't studied this particular area have limited understanding of how it works.

In this section, we aim to explain a few key ideas behind neural networks, and use them to answer this question: how can sensitive data of a person's driving history be used to train an autonomous driving agent without retaining the data itself? A neural network is a particular type of mathematical function that can be implemented efficiently on modern computer hardware. Input data is fed into the network in what is called the "input layer" of the network. Different kinds of "layers" of mathematical transformations are applied to this data, with each layer learning a higher-level representation of the data. The most important kinds of layers including applying an affine transformation to the data of the form $y = Wx + b$ and applying nonlinear "activation functions" such as a rectified linear unit (ReLU) $f(x) = \max(x, 0)$ or a hyperbolic tangent $f(x) = \tanh(\mathrm{x})$. A convolutional filter applies a set of weights to a local grid, often pixels, and shares these weights across the whole image. Convolutional Neural Networks (CNNs) have been particularly successful in image recognition tasks. "Deep" networks are those with more than a handful of layers. Deep convolutional networks are deep networks with multiple convolutional layers. This architecture in particular has been effective at computer vision tasks.

A complete introduction to neural networks is beyond the scope of this paper. Numerous textbooks and online tutorials are available for the interested reader. Andrew Ng's online courses offered at Coursera and DeepLearning.ai are free, highly accessible and recommended to the interested reader.

How can a neural network learn a useful driving task, such as recognizing objects, without retaining the data it observed? This is actually a very intuitive notion, because humans do it all the time! We can distinguish humans from other images with tremendous efficiency even though we hardly remember every person we've ever seen. The ability to recognize a person is encoded in the neural pathways of our brain even though the specific images used to develop that capability have been lost. Neural networks have the same high-level properties. The capabilities that the network has "learned" are encoded in the parameter weights of the network. For example, an affine layer $y = Wx + b$ converts an *n* dimensional vector of inputs x to an *m* dimensional vector of outputs y. The matrix W has shape *m* x *n*, and each of these *mn* entries is an unknown parameter weight. The vector *b* similarly contains *m* unknowns.

Neural networks are most commonly trained using a technique called gradient descent. The network designer specifies a loss function, and the gradient descent seeks to minimize this function by using a local estimate of the derivative of the loss function with respect to the weights. One classical example is classifying hand written digits. The network is presented with an image corresponding to a known digit 0-9. It outputs estimated probabilities that the image is each digit. The cross-entropy loss function is the negative log of the probability assigned to the true class. This rewards the network for assigning a high probability to the correct class.

The technique of backpropagation is used to efficiently compute the gradient of the loss function with respect to all of the parameter weights in the neural network. The largest neural networks that win competitions such as ImageNet can have on the order of hundreds of millions of parameter values. In gradient descent, after each batch of training data is analyzed, the network is updated to take a small step in the direction that would minimize the loss function. The size of this step is determined by the gradient and the so-called learning rate parameter.

The key idea here is that the weights embody everything that the network has learned about the task. Once a batch of data points have been used to update the network weights, the data can be discarded and the network will still retain whatever it has learned. While the *network* does not require retaining the data to run, non-privacy-preserving deep learning still requires that data owners trust the model builders, as these humans have access to the potentially sensitive data. In the sections below, we will explain how new techniques in training neural networks could be used so that vehicles could train neural networks to perform tasks related to autonomous driving without compromising the privacy of the underlying training data. Simply put, a vehicle could be equipped with a system that could train an autonomous agent to drive based on the data generated by the vehicle, while preserving the privacy of the human driver.

Federated Learning and Distributed Training of Neural Networks

The first concept we introduce is termed Federated Learning (FL) and is already widely applied in the tech industry to enable services such as the next-word prediction on smartphones; in this system, software providers train a next-word prediction model without seeing your messaging history. The basic idea is that to train a neural network we only need the gradients—a vector which gives us the direction to move in our optimization procedure—and nothing more. FL proceeds as follows:

1. A model owner sends their model (specified as a program) to the data owner (e.g. a vehicle equipped with appropriate sensors);
2. The data owner allows the program to be executed on their latest data; the program returns the gradients of the computation (the specific logic for this computation is specified by the model owner);
3. The data owner performs this computation on a batch of data and aggregates the gradients, or even better, applies a small stochastic perturbation to the weights (differential privacy). Both of these procedures provide security by reducing the chance that an adversary can reconstruct the input data from the gradients;
4. Having received the gradients from the data owner, the model owner updates their model weights; the process repeats.

By following this procedure for many mini-batches of data and distributing this computation across many data owners (e.g. millions of cars, smartphones, etc.), the model owner can obtain

a very effective model without ever “seeing” the data it used. We present a diagram to illustrate this point below.

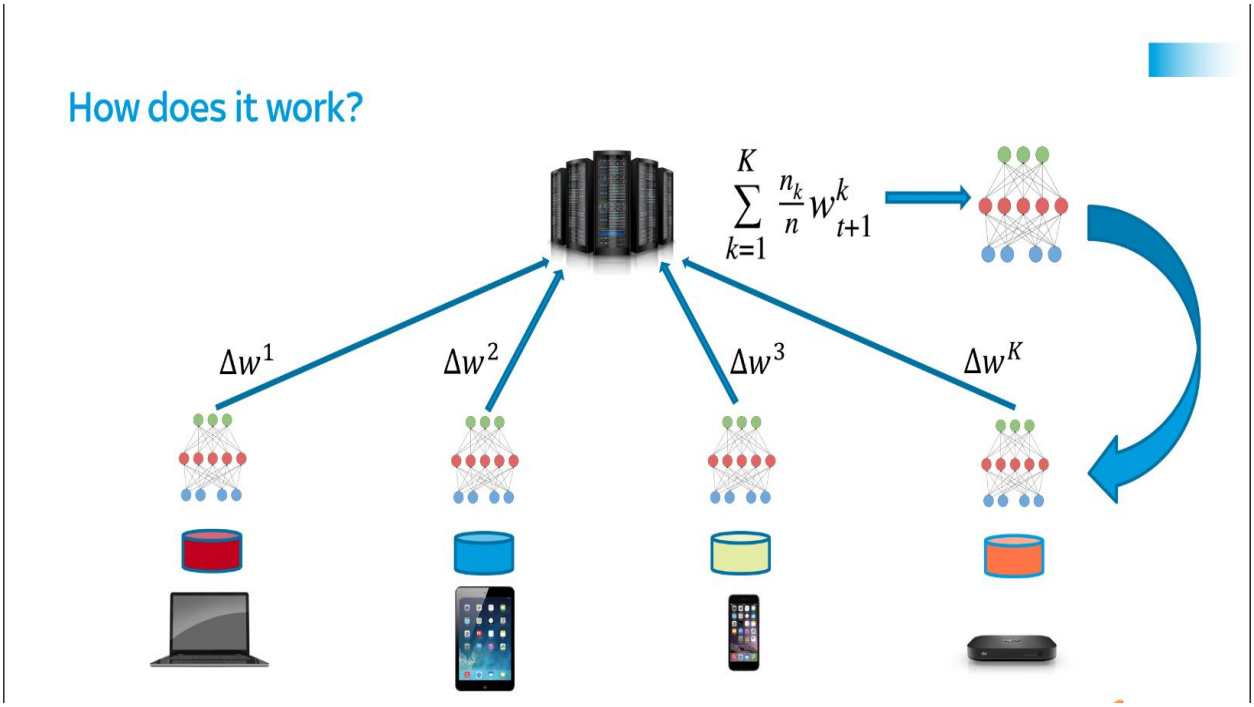

Figure 1: Federated learning in action

As mentioned above, this scheme is already deployed on a massive scale in the smartphone industry. We believe that the autonomous vehicle industry could unlock extremely valuable driving data from the billions of miles driven each day by following this Federated Learning paradigm.

Secure Multi-Party Computation

To truly unlock driving data stored *anywhere*, however, we need a way to protect the privacy of the model owner too. If Audi wants to train a model in a FL manner from its own vehicles, it can directly protect its model as it also develops the operating system deployed on the car. But if Audi would like to collect data from Mercedes vehicles, there would be no guarantee that an adversary wouldn’t steal their valuable model architecture and weights.

This is where Secure Multi-Party Computation (SMPC) comes into play. The idea behind SMPC is to randomly split sensitive numbers into “shards”, encrypt the shards and perform computation on them, and retrieve the unencrypted result only if all parties involved in the computation re-contribute their shard. By doing this, a model owner can train their model without revealing the true values of the weights of their neural network; and it is the weights which are

truly the valuable component. We demonstrate this basic yet profoundly powerful procedure below.

```
Q = 1234567891011  # large-enough number
x = 25             # number we wish to distribute for computation

import random

def encrypt(x):
    share_a = random.randint(0,Q)
    share_b = random.randint(0,Q)
    share_c = (x - share_a - share_b) % Q
    return (share_a, share_b,  share_c)

encrypt(x)
>>> (267553417101, 442119224293, 524895249642)

def decrypt(*shares):          # decryption requires all shares to be
    return sum(shares) % Q     # provided, otherwise result remain encrypted

decrypt(267553417101, 442119224293, 524895249642)
>>> 25

decrypt(267553417101, 442119224293)
>>> 709672641394
```

Figure 2: Encrypting data with random shards

Above we see the basic *encrypt* and *decrypt* functions. We see that if we do not provide all three shards to *decrypt*, the number returned is still encrypted. Next, we demonstrate the truly amazing piece: performing computation over encrypted values.

```
x = encrypt(25)
y = encrypt(5)

def add(x, y):
    z = list()
    # the first worker adds their shares together
    z.append((x[0] + y[0]) % Q)

    # the second worker adds their shares together
    z.append((x[1] + y[1]) % Q)

    # the third worker adds their shares together
    z.append((x[2] + y[2]) % Q)

    return z

decrypt(*add(x,y))
>>> 30
```

Figure 3: Addition with encrypted numbers

Combining addition with multiplication and comparison functions, we can implement all possible functions required in neural network computations. Thus, this scheme allows us to protect model owners when distributing their computation in federated learning.

Recently a community of developers has gathered to write tools on top of PyTorch for performing these operations. We would be remiss not to acknowledge https://github.com/OpenMined/PySyft for their fantastic library and tutorials.

Performance Benchmarks: Training a Simple Neural Net Two ways

Of course, there's no such thing as a free lunch. Utilizing these various privacy-preserving tools for deep learning incurs a cost in terms of overheads during training. To demonstrate these, we performed experiments with the PySyft library in which we trained a private, federated convolutional neural network model on the MNIST digit classification dataset. We computed mean ping times between our machines and various AWS servers (Ohio, Northern Virginia, California, etc.) and randomly sampled these times to add as latency overheads during training. This way we simulate the transfer of model and gradient data which would be present in a real-world deployment. Below we present the results of our experiment.

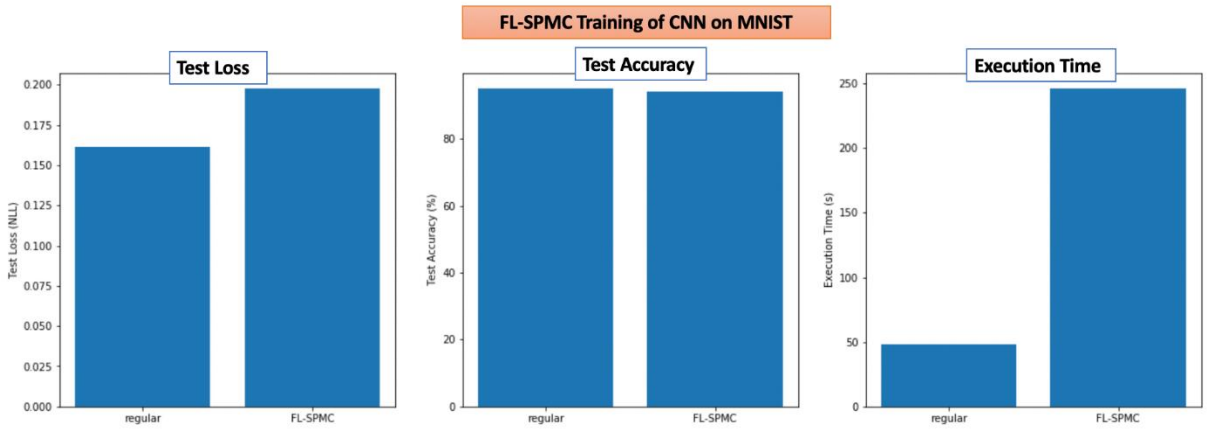

Figure 4: Test loss, test accuracy and training execution time for FL-SPMC on MNIST with simulated latency

We see that the FL-SPMC model obtains roughly equal accuracy and a slightly higher loss, with around 5x higher execution time. Without the simulated latency, execution time is roughly 2x higher.

In conclusion, we see that it is certainly possible to leverage the billions of miles driven daily to train neural networks for autonomous vehicles while providing security and privacy for data and model owners. There are overheads associated with the communication inherent in federated learning and in the computation required for secure multi-party computation, but we believe these are worth the pain for greatly increased quantities of data which hold the promise of taming the long-tail of unlikely events that AVs must handle to be deployed in the real-world.

Current Leaders in Autonomous Vehicle Technology

As Bloomberg News puts it, "in the race to start the world's first driving business without human drivers, everyone is chasing Alphabet Inc's Waymo."[5] California has required detailed disclosures of all accidents and "disengagements" involving autonomous vehicles since 2014. A disengagement is when a human safety driver intervenes to take over driving the vehicle. Industry experts have used the California statistics to assess the strength of different competitors. Waymo has by far the lowest disengagement rate, covering 11,154 miles per

[5] Bloomberg News Who is Winning the Self Driving Car Race?

disengagement. GM Cruise is a clear second place with 5,205 miles per disengagement, with the rest of the field below 2,000 miles.[6]

The miles per collision showed the same leaders. Waymo was in first place with 350,000 miles covered and 3 crashes for a mean distance of about 117,000 miles between crashes. GM Cruise had 22 crashes over 132,000 miles for a mean distance of 6,000 miles per crash. It should be noted that California officials deemed the autonomous vehicles "not at fault" in almost all of these collisions. At the same time, the AVs sometimes behave in legal but unpredictable ways, such as jamming on the brakes more abruptly than a person would to avoid perceived obstacles that may not be real. This can lead to more accidents in which people were "at fault" for rear ending the AV, but which wouldn't have happened had the AVs been smoother.

Waymo: Slow and Steady Wins the Race?

Waymo is a division of Alphabet, the parent company of Google. The name is intended to suggest a way forward for mobility. It has been developing autonomous vehicles since 2009, when the company (still called Google at that time) launched "Project Chauffer." Waymo has taken a consistently gradual and cautious approach to rolling out its vehicles. Before launching the world's first commercial autonomous vehicle service in the Phoenix, Arizona area in the middle of 2018, Waymo offered free rides to 400 "early riders" from the start of 2017, and offered rides without safety drivers starting in October 2017. Waymo gathered extensive feedback and built a track record of safe operation before launching the service commercially on a small scale. Waymo carefully chose to start in Arizona because of its dry climate and traffic layouts that were considered to be among the easiest in the US.

To put Waymo's accident rate in context, there were 12.8 million vehicles involved in car accidents in the US in 2016, implying a rate of approximately 250,000 miles driven per accident.[7] We saw above that Waymo vehicles tested in California covered an average of 117,000 miles per collision. This would tend to suggest that Waymo cars are approaching an accident rate that is comparable to human operated vehicles, but still get into more crashes (disregarding at the moment any notion of which operator if any was "at fault" in a collision). It is likely that Waymo is reporting less serious accidents than the overall US rate, which is based on NHTSA statistics. As a practical matter, car accidents in the US are only reported to authorities if someone is injured or if an insurance claim is made. Many minor "fender benders" are ignored or settled privately between drivers to avoid the hassle going through insurance and seeing possible premium increases. Waymo on the other hand is under a microscope and reports even the most minor incident in California authorities. Our estimate is that Waymo vehicles operating in California today are roughly comparable in safety to vehicles operated by people.

Waymo has emphasized software simulation as a major component of its development program. It has developed a car simulation program called Carcraft (named after *World of Warcraft*) and it runs nonstop, continuous testing on a fleet of 25,000 virtual autonomous vehicles.[8] While Waymo's fleet has covered over 10 million miles in the real world as of late 2018, this virtual fleet had covered over 10 *billion* miles. Drago Anguelov, the principal scientist at Waymo quoted earlier, said in his guest lecture at MIT that he viewed virtual testing as the

[6] California AV Disengagements 2018
[7] US Car Accidents in 2016
[8] Wikipedia - Waymo Also verified in Drago Anguelov's lecture at MIT

primary means by which Waymo improved its algorithms and performance. His view is that road testing was intended to validate their simulator testing, not to learn new things about the world in the first instance. Waymo has embraced a "safety first" culture from the beginning, and repeatedly and publicly stated that safety is their top priority in developing autonomous vehicles.[9]

The Culture at Uber and the Curious Case of Anthony Levandowski

Early 2017 was a difficult time for Uber. The company's reputation was in the process of metastasizing from the bold, brash provider of an irresistibly convenient service to poster child for the problems of companies led by arrogant and immature "tech bros." The company had long faced criticism that it exploited its drivers by paying less than a living wage, and been very aggressive about flouting inconvenient government regulations on taxis. But a rumble became a roar when a few high profile stories broke in the space of two months.

Travis Kalanick's reputation as a bad-boy was long-standing and well earned before he was caught on video shouting at an Uber driver who was complaining he was unable to earn enough to support himself.[10] Kalanick told him "Some people don't like to take responsibility for their own sh*t". Kalanick's previously most prominent arrogant gaffe had been calling Uber "boob-er" for facilitating his … romantic liaisons… with young women. More consequentially, an engineer at Uber named Susan Fowler brought the company to its knees with an incisive blog post[11] that shredded the company and its culture for pervasive sexual harassment. It was an early step in the me-too movement, and eventually led Kalanick's ouster.

Against this backdrop, the lawsuit filed by Waymo against Uber for an alleged theft of intellectual property was catnip for the press and all the Uber-haters in the world, and we got to learn about the strange world of Anthony Levandowski. Levandowski had been one of the leaders of Google's Project Chauffeur before Kalanick and Uber poached him. Uber paid a staggering $600 million for a six-month old startup OttoMotto that Levandowski started after leaving Google. Uber extended him a term sheet just a month after he started the new company. The widespread perception was that they paid not for anything developed by Otto, but for Levandowski and a trove of Google intellectual property that he took with him.

The New Yorker wrote a gripping full-length feature about Levandowski and the lawsuit. This and other articles revealed that Levandowski had downloaded 14,000 documents from a Google server, copied them to an external drive, then wiped his laptop. He and some other Google leavers had shared text messages about deleting files and message history that made them sound like illegal conspirators. A Google employee was accidentally CC'd on an email from a vendor showing that Uber had ordered a microchip used for lidar with an essentially identical lithography to a proprietary chip designed by Google. That's when Google decided to sue. The lawsuit was eventually settled[12] with Uber granting Waymo a 0.33% equity interest in Uber, which was valued at approximately $245m as of the settlement. Uber also agreed not to use any of the stolen technology.

---

[9] Below we will see how tension over the priority of safety vs. rapid development led to the departure of Anthony Levandowski from Waymo to Uber

[10] Bloomberg Kalanick Argues with Uber Driver

[11] Susan Fowler's Blog Post about Uber Wikipedia - Susan Fowler

[12] NY Times Waymo vs. Uber Lawsuit Settled

The most relevant aspects of the Levandowski story though for this discussion about the safe development of autonomous vehicles are the myriad anecdotes that illustrate his values and character. Our argument is that if you were a sane executive in charge of developing autonomous vehicles, you would not want a person with these views and behavior patterns in a position of authority. The most striking claim was this quote attributed to Levandowski: **"I'm pissed we didn't have the first death."** [13] The context of this alleged remark (denied by Levandowski) is that Uber was too conservative about rolling out new technology. The implication is that Tesla (which had the first two fatalities in semi-autonomous operation) was more of a risk taker and would be rewarded. It's worth noting that this article came out before Uber "achieved" the first pedestrian fatality.

Whether or not Levandowski advocated racking up a body count at Uber, other quotes are not in dispute. This is a quote from the New Yorker article about Levandowski's time at Google:

"Some of the biggest fights involved risks that Levandoswki was taking in self-driving experiments. The software that guided Google's autonomous vehicles how to, say, merge onto a busy freeway is to have them do so repeatedly, allowing their algorithms to explore various approaches." Levandowski said in the interview "if it is your job to advance technology, **safety can never be your No. 1 concern**. If it is, you'll never do anything. **It's always safer to leave the car in the driveway**. You'll never learn from a real mistake." This is a recurring theme in Levandowski's statements and emails. Emails released in the trial included quotes such as "The team is not moving fast enough due to a combination of risk aversion and lack of urgency, we need to move faster" (Levandowski to Larry Page at Google) and **"Second place is first looser [sic]"** (Levandowski to Kalanick at Uber.)[14]

Back in 2011, another executive at Google, named Isaac Taylor, went off on paternity leave and found out on his return that Levandowski had surreptitiously disabled a system that forbid the cars from taking routes deemed too dangerous. When Taylor found about this, he got into a shouting match with Levandowski. To persuade Taylor of his position, Levandowski induced him to jump into a car together to take one of the forbidden routes. Ironically, on this single demonstration trip, the Waymo vehicle aggressively cut off a Toyota Camry on the highway, forcing it onto the shoulder as it took evasive action. The Waymo vehicle ended up in a serious crash, injuring Taylor to the point he required multiple surgeries to repair damage to his spine.

Then there is the straight-up weird stuff about Levandowski. At the time of the lawsuit, he was in the news for promoting a church called Way of the Future, devoted to "the realization, acceptance, and worship of a Godhead based on Artificial Intelligence." Early in his career, Levandowski had a colleague who he thought wasn't working hard enough because he was "distracted" by a romantic relationship. So Levandowski offered the man's girlfriend $5,000 to break up with him. She declined the offer, and they lived happily ever after (well, they got married anyway), and the engineer doesn't work with Levandowski anymore.

The Death of Elaine Herzberg, First Pedestrian Killed by a Robot Car

Uber is not the second, third, or fourth ranked competitor on anyone's list of leaders in the autonomous vehicle industry. Why are we even writing about Uber in this paper? Because the company holds the dubious distinction of operating the first autonomous vehicle to kill a pedestrian. On March 18, 2018, one of Uber's test vehicles struck and killed a pedestrian

[13] New York Magazine - Is Uber Doomed?
[14] TheVerge - Levandowski Put Safety Second

named Elaine Herzberg in Tempe, Arizona. Herzberg was crossing the street outside of a legal cross walk, pushing a bicycle with grocery bags. It was 9:58 PM and dark outside. The Uber was traveling at 43 mph, under the legal speed limit of 45 mph. But the software on the Uber failed catastrophically. Herzberg appeared on the sensor data 6.0 seconds / 378 feet before she was struck.[15] The car at that speed would have needed 196 feet to stop. Over the next 4.0 seconds, the system did not detect that emergency braking was required. It classified Herzberg first as an "unknown object", then as a vehicle, then as a bicycle. Each time, it updated her likely trajectory. The path planning module in place at that time had no concept of uncertainty or braking because the potential path of the object was unknown and might lead to a collision. It was deterministic and only called for braking when the one projected path led to an obstacle.

The system finally determined that emergency braking was required 1.3 seconds and 76 feet before impact. The safety driver in the car, Rafaela Vasquez, did not have her eyes on the road. A police investigation concluded she had been watching the T.V. show "The View" on her phone using the Hulu app. Uber engineers had developed an emergency braking system that would engage the brakes when the system determined it was necessary. But other Uber executives had decided to disable this feature because the vehicles stopped too abruptly when it was on. They did not however take any action to lower the speeds taken by the vehicle. So when the system determined emergency braking was required, rather than braking, it alerted the distracted safety driver. She first touched the steering wheel 1.0 seconds before impact, and only applied the brake one second *after* the impact

In the aftermath of the accident, Uber faced a torrent of negative publicity. They immediately canceled all of their autonomous vehicle trials. Nine months later, they resumed testing in Pittsburgh on a smaller scale, and don't have any plans to return to Arizona as of now. Uber settled two lawsuits with members of Herzberg's family under confidential terms. It is speculated that they paid out millions of dollars. Of more consequence, Uber's cavalier approach to safety has badly damaged the reputation of the entire industry, and made the public more skeptical of autonomous vehicles and resistant to their introduction.

The Importance of Company Culture

What is the connection of all these anecdotes, fascinating though they may be, to the course on Critical Thinking in Data Science we have just taken? If we could distill the essence of the course into one sentence, we would channel Aretha Franklin:

You better think (think)
Think about what you're trying to do to me

As technologists, many of us in the class are going to go out into the world and get jobs working on products and services. These products and services have an impact on people. We need to think critically about how our decisions and actions as engineers are going affect other people.

One recurring theme in the course has been that leaders of technology companies have sometimes approached their actions in purely technical terms without pausing to consider potential or even likely consequences of their decisions. Probably the most prominent example discussed in the class has been Mark Zuckerberg and Facebook, which blithely asserted that "connecting people" was an unambiguous good for the world and proceeded to hoover up

[15] See Wikipedia - Death of Elaine Herzberg and N.Y. Times: Self Driving Uber Kills Pedestrian for a detailed description of exactly how the accident unfolded and the various failures of the Uber vehicle.

massive quantities of private data for years with minimal reflection about how their actions impacted Facebook users and the wider community.

Towards the end of the course, in our final group discussion, we were urged to always ask the question **"what is your role"** as a responsible technologist. While there is no one right answer, we claim that there are some answers that are clearly wrong. Without equivocation, using only one hand, we will come right out and say Anthony Levandowski's approach is the wrong way to develop autonomous vehicles. We will paraphrase his implied argument to be that autonomous vehicles will save lives, so we need to push to get them out as fast as we can. If a few people get killed, that is part of the price of progress. We reject this argument.

Another recurring theme in AC 221 has been that while ethical questions usually don't have clear answers, they often reflect our values, and those values are often direct. We hold the values that engineered products (whether hardware or software) should be made as safe as is feasible. If we are working on an autonomous driving system, it is our responsibility to make it as safe as it can possibly be with the set of tools we are using. If that means a test deployment needs to be delayed, so be it. It's easy to hold up Levandowski as an example of values we disagree with, but he is just one man who worked in bigger organizations. All the other people in those organizations made their decisions too, and ultimately it is clear that each organization has its own values that is an aggregate of the values of the individuals who work there. At Google, Levandowski was surrounded by engineers like Taylor and Waymo CEO John Krafcik who viewed safety as their number one priority. At Uber, rank and file engineers made the fateful decision to disengage the emergency braking system and instead have it signal the safety driver to stop.

We can analyze this one decision to disengage the emergency braking system as being driven primarily by values, not engineering. The engineering decision was made to activate emergency braking once a minimum safe distance in front the car was no longer assured. This was unpopular, because the ride was too jerky, with too many sudden stops. At the risk of stating the obvious, at a company with a safety culture, engineers wouldn't have turned off the emergency braking feature, they would have reassessed the situation. If the car was stopping so aggressively, maybe it simply wasn't ready for this trial yet? GM Cruise has limited its vehicles to speeds of 25 mph for years, because they believe it isn't ready to operate safely at higher speeds. Google, as we know from the New Yorker article, didn't permit its vehicles onto highways in 2011 because it thought they weren't ready yet. Individual software engineers have a choice and a voice. If someone tells us to do something dangerous, we can stand tall and just say no.

There will always be a tension between commercial and competitive pressures and values such as safety or privacy. In the brief duration of this course, we've seen another famous engineering company enmeshed in a crisis because it cut corners on safety to catch up to a rival: Boeing. Two Boeing 737 Max-800 aircraft crashed in a period of four months, killing 346 people, due to flaws in a software system called MCAS[16]. Without rehashing the whole debacle, suffice it to say that the parallels are obvious. Each company (Uber and Boeing) feared it had fallen behind rivals (Waymo and Airbus) in a race to produce a lucrative new

[16] https://en.wikipedia.org/wiki/Boeing_737_MAX_groundings

product. Engineers at each company who raised safety concerns were overruled in a rush to ship a product, leading to a tragic and avoidable loss of life.

How Much Risk is Acceptable in Deploying Autonomous Vehicles?

We have argued above that Uber has taken unacceptable risks in deploying autonomous vehicles before they were ready. We are emphatically not arguing however that the right amount of risk is zero. If we were to wait for a perfect autonomous vehicle, we will probably never get one, and will continue to suffer 1.25 million traffic fatalities annually around the world. On a smaller scale, every parent (up until now anyway) has been faced with the quandary of their teenage child wanting a driver's license. Their collective experience strongly suggests that forbidding teenage kids to learn to drive is not a viable parenting strategy.

We will now apply another core tenet learned in AC 221: these questions don't have a single right answer. There are a range of reasonable approaches and they reflect the values of those making the decisions. Our values suggest that a sensible level of risk should be based on two considerations: a numerical estimate of safety and a required degree of diligence.

The safety of an autonomous vehicle should not be measured only by a grim total of fatalities. Fatalities are too rare and too costly. The approach taken by California regulators requiring reporting of accidents and disengagements makes it feasible to assess the overall safety profile of different autonomous vehicles being tested in that state. A simple and direct guideline is that autonomous vehicles should not be tested without safety drivers unless they are operating on a par to human operators. The scale of a test deployment should be tied to their relative performance. We argued above that after accounting for the difference in severity for accidents reported to the NHTSA and California, Waymo vehicles have a plausible claim to be roughly on a par with human operators for safety. GM Cruise vehicles are not yet on a par with Waymo, but they are being tested responsibly inasmuch as they are always tested with safety drivers and limit speed to 25 mph.

The second key criterion is degree of diligence. Simply put, an autonomous vehicle should not be tested on public roads unless it is as good as it can reasonably be with the existing generation of technology. This doesn't mean perfect, but it does mean that clear problems must be addressed in an earlier testing stage before vehicles are unleashed on public roads. One of the safest ways to improve autonomous vehicles is to emphasize simulated testing (i.e. in computer simulations) over road testing. This is an approach taken by both Waymo and Cruise. Simulations are not yet capable of revealing every risk of course. When problems arise in testing, companies need to be responsible about limiting the scope and degree of difficulty in a test deployment. Accidents like the Uber crash didn't happen in a vacuum; there were many close calls that had been disregarded before Herzberg was killed.

We would like to tackle head on the argument that deploying AVs this slowly will lead to too many lives lost to avoidable accidents caused by human drivers. This argument is predicated on the assumption that people would accept an early widespread release of AVs at a time they were only comparably as safe as human operated vehicles. The "pot of gold at the end of the rainbow" would come only later, after improvements made them safer. But this assumption is belied by extensive evidence of how people behave. Simply put, we hold machines to a higher standard than we hold ourselves. Is this rational? Is it right? We hold that these questions are beyond the scope of this paper, and in any case largely irrelevant, because people are unlikely to change.

In the aftermath of the Uber pedestrian fatality, a poll taken by AAA suggested that 73% of American drivers would be afraid to ride in an autonomous vehicle, up from 63% in October.[17] This is not an effect limited to older drivers who might be techno-phobic; 64% of millennials age 20-37 were also afraid of AVs. In this opinion environment, it strikes us as naïve to think that Americans will accept an AV deployment in which hundreds or thousands of Americans are killed without a demonstration that the AVs are overwhelmingly safer than human operated vehicles. If we were forced to speculate on how much safer they would need to be, we would say at a minimum twice as safe. More realistically Americans might insist that AVs were a full order of magnitude (10x) safer before accepting their widespread deployment. A comparison with airplane fatality rates is sobering: people seem to be highly averse to plane crashes, much more so than car crashes. In recent years, the fatality rate for air travel in the US has been approximately 0.2 deaths per 10 billion passenger miles.[18] The same statistic for cars is approximately 150 deaths per 10 billion passenger miles, indicating that air travel is 750 times safer than car travel. Yet every plane accident is a national news story, while an auto fatality is local news if it is covered at all unless the victim was famous. This is the backdrop leading us to speculate that the politically acceptable safety performance of AVs is potentially a full order of magnitude safer than human operated vehicles.

What is the Optimal Regulatory Framework for Autonomous Vehicles?

The discussion above was addressed to choices we suggest that companies make on a voluntary basis about the right tradeoff between safety and speed in testing and deploying autonomous vehicles. Governments face a closely related question about how (and even whether) to regulate autonomous vehicles. Some jurisdictions have almost no regulations on autonomous vehicles. Arizona is one of them, which has been a major reason it has been a site for trials along with its dry weather and traffic patterns.

California has historically taken a more proactive approach to regulation generally. Opinions about California's regulatory climate often reflect partisan politics in the US, with liberals supporting a greater government role and conservatives criticizing the state as a place that is hostile to businesses and commercially successful in spite of its regulatory overreach. We would like to avoid partisan politics here and narrowly focus on California as a test case for the regulation of autonomous vehicles.

Dating back to 2014, California has required permits for companies that wish to test autonomous vehicles. California has also required that companies report total miles traveled, disengagements, and accidents. While some may have predicted that these regulations would make it unattractive to test autonomous vehicles in California, it hasn't worked out that way, and the state is a hotbed of activity in the industry. A look at the "league tables" of industry leaders according to consulting and investment research firms often looks suspiciously similar to a list of the companies testing in California sorted by total miles driven or average miles per disengagement. In that respect, the California regulations have helped the industry to measure its performance and gain the trust of a wary public. Knowing where the bar set by the leaders has also pushed the rest of field to improve, and encouraged them to pace their own test deployments responsibly.

[17] CNBC - Self-Driving Cars are Scaring More People
[18] https://en.wikipedia.org/wiki/Transportation_safety_in_the_United_States

Before autonomous vehicles can shift from testing to deployment, society will also need to address the question of legal liability. This is an open issue without a clear answer, and a number of approaches might prove viable. We suggest that a sensible approach would be to continue with minimal changes to the existing system of liability insurance. All states mandate minimum levels of liability coverage to obtain a driver's license. To own an autonomous vehicle legally, states could similarly require liability coverage. The argument to go through private insurance companies is that they are large sophisticated corporate entities with commercial power comparable to companies that produce AVs, and they have an expertise an assessing risk. They should be well suited to create market forces that will reward companies for manufacturing safe vehicles and penalize those that manufacture risky vehicles. Using private insurance will also allow companies to price differentially so that people who drive their cars more, and on more dangerous roads, will pay more. This will create another incentive to discourage risky behavior, even if the risk in question is sending your robot car on a long and slightly dangerous daily commute.

Putting it Together: Private Data Plus Massive Simulation for Safer Autonomous Vehicles

If a perfect autonomous vehicle could be developed, the world could avoid 1.25 million traffic fatalities a year. But the development of an autonomous driving system even on a par with a human operator remains a "grand challenge"—perhaps *the* grand challenge—of artificial intelligence and engineering in the 21st century. We have argued that people will only accept a widespread deployment of AVs if they are much safer than human operators. While this preference may be irrational, it is clearly demonstrated and unlikely to change. This leads to a chicken and egg problem: how can the industry develop a system with superhuman performance if it can't test and deploy on a wide scale?

Industry leaders including Waymo and GM Cruise are showing the way forward. These systems will be developed on a foundation of massive computer simulations. The primary function of real-world testing at Waymo is already to validate computer simulations and persuade the rest of us that their vehicles are safe. But designing a simulation with sufficient verisimilitude to capture a serious accident is dramatically more difficult than simulating ordinary traffic conditions, and it remains an open challenge. This is the "long tail" of driving that is the focus of Waymo's simulation efforts.

The one fatality so far is a prime example of this long tail of complexity. The pedestrian who was killed was pushing a bicycle outside of a cross walk late at night. A toxicology report also indicated the presence of both methamphetamine and marijuana in her blood. While this doesn't prove she was high on drugs when she was hit, it is quite likely that her reactions were impaired.

We argue that one promising approach to tame this long tail of complex and exceedingly rare events is to obtain massive amounts of real-world data that can be used to train neural networks and simulate traffic conditions. While people have been surprisingly willing to part with their private data on social networks, they have shown far greater reluctance to reveal their driving history. We believe that evolving techniques in Federated Learning, Secure Multiparty Communications, and Differential Privacy could be transformational. A blend of these technologies, if properly configured, could allows millions of drivers to use their driving data to train autonomous driving systems without surrendering their privacy. That is a future we would like to see.