Advances in Cognitive Systems X (20XX) 1-6 Submitted X/20XX; published X/20XX

Anticipatory Thinking Assessment: Stress tests using synthetic data

Adam Amos-Binks AAMOSBINKS@ARA.COM
Leilani Gilpin LGILPIN@UCSC.EDU

Abstract

As autonomous vehicle begin to make human-level decisions, users, stakeholders and system experts will need to be confident in their decisions and behaviors. One way to ensure that these systems are robust is to measure their ability of anticipatory thinking; or how well they can reason about out-of-domain situations. One way to test for these new situations is to design stress tests, or iterative testing procedures that intend to perceive the agents faults under successive pressure. This procedures "break downs" the component to failure, so system debuggers and auditors can evaluate the strength of the mechanism being tested. We show that you can apply the same procedure to perception systems. We should how you can apply different mask to perception data "break down" the perception system to predict different objects. By combining this with a knowledge graph of information, we can use these approaches to start measuring anticipatory thinking in safety-critical and mission-critical AI systems.

1. Introduction

Anticipatory thinking drives our ability to manage risk from the everyday to expert-level performance. Advances across AI has led to deployments in safety-critical and mission-critical domains like medical diagnosis and autonomous driving where managing risk is essential. As machine learning continues to set new performance standards for the field, AI systems will become more pervasive in these domains.

However, there are still many real-world examples of AI systems mismanaging risks leading to errors that humans would not make. Without any knowledge of how these systems manage risk, we are limited in their application to safety and mission critical domains. The opacity to managing risk is in part due to typical task-based performance measures (e.g. error rates on static tests) that limit transparency of future performance in plausible everyday scenarios that are nonetheless outside the normal testing range. Calls to develop ability-based assessments that build on human abilities are still fledgling, with concepts such as universal psychometrics in their infancy (Hernández-Orallo, 2017). Adversarial robustness efforts to harden machine learning models against actors with malicious intent are advancing but neglect natural adversarial examples that remain intractable to enumerate and test in advance. Without a transparent and semantic anticipatory thinking assessment, there is self-imposed ceiling on the risk management capabilities of AI systems, the effects of which are news worthy mistakes because they could have been avoided.

Our ability-based assessment elicits an AI system's anticipatory thinking, revealing how it manages risks with respect to types of human decision making across tasks. Impacts of using our as-

sessment include a more accurate characterization of real-world deployment performance, sparing casualties in safety and mission critical domains. We accomplish this with three contributions:

1. Synthetic data collection
2. Anticipatory Thinking assessment
3. Working prototype with evaluation results

Our first contribution is the counterfactual synthetic data generation. We generate a set of synthetic data images by modifying observed data in plausible and realistic ways. The outut of this process are a set of counterfactual data. Counterfactuals reveal how an AI system manages risk in 'what-if' scenarios that include rare and long-tail examples.

Our second contribution is the anticipatory thinking assessment for an existing image classifier. We include formal definitions of each component of the assessment in Section 3. These definitions are essential to generalize the assessment beyond the running example and toy domain we refer to. This makes our assessment adaptable to other types of stress testing problems beyond stop signs in self-driving cars. A workflow of our assessment is in Figure 1.

Finally, we provide preliminary results on an example. We create a set of synthetic stop sign data for a self-driving car. We apply our anticipatory thinking assessment on our dataset and an existing model, ResNet-50. These results show that our approach can be used for practical applications and that many of the components needed are readily availabl. Our work can be extended to many state of the art/practice tools such as knowledge graph embeddings and GANs.

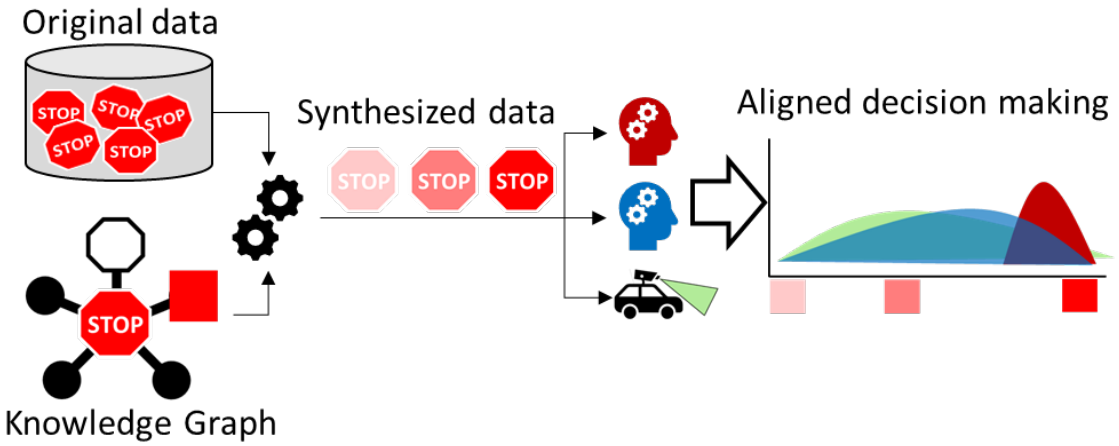

Figure 1: We generate plausible, synthetic data to reveal hidden failure modes in AI systems when compared to human decision making. Our synthetic data is the result of reasoning over a knowledge graph and advancements in generative AI.

(a) An example of an object-level perception error. (Credit to `_realrusty` on TikTok.)

(b) An example of a scene-level perception error. (Credit to `u/VREsports` on Reddit.)

Figure 2: The two types of anticipatory thinking autonomous vehicle perception problems. The object-level errors occur when the object classification is wrong, .e.g., the moon being mistaken for a yellow light. The scene level classifications occur when the objects in the scene is correct identified, but the scene understanding is incorrect: a truck full of traffic lights.

2. Background and Related Work

Our contributions build upon previous work on anticipatory thinking (Amos-Binks & Dannenhauer, 2019) in open world domains (Amos-Binks et al., 2021), robustness, and stress testing AI systems.

2.1 Anticipatory Thinking for Autonomous Systems

Anticipatory thinking (Geden et al., 2019) is behind our ability to manage risk in real world domains like driving. The problem is that while humans are able to anticipate rare events in driving, an autonomous vehicle cannot. For example, if human drivers perceive a skateboard rolling across the street, they may intentionally slow to a stop for children, parents, and friends chasing it.

On the contrary, autonomous vehicles are perceiving the world in a different way. Autonomous vehicles are constantly perceiving their surrounding environments with perception systems, e.g., vision, LiDAR, and radar, to resolve the difference between reality and their representation of it. The vision system, which is opaque to humans, is not prepared for rare but highly impactful perception errors; making autonomous vehicles an open-world domain Langley (2020).

Self-driving has been proposed as an open-world anticipatory thinking perception challenge (Amos-Binks et al., 2021). There are two types of challenges, one at the object level, and one of the scene level. Object level challenges include perceptions where the objects identified are clearly wrong, e.g., a mailbox crossing the street (Gilpin et al., 2018), or the moon being perceived as a yellow light in Figure 3a. Whereas the scene level challenge occurs when the objects are correctly identified, but there is a lack of scene understanding. E.g., a truck carry traffic lights, but the perception system does not understand that the traffic lights are not "active." The illuminates an issue with scene

understanding: low-level objects and their behaviors are memorized instead of being understood in different environments.

2.2 Human Alignment and Robustness

When AI systems fail, they fail in different ways than their human counterparts. For example, AI systems lack commonsense (Davis & Marcus, 2015). These autonomous agents are increasingly deployed in real world settings, where there has been an increase in malfunctions and errors leading to injuries[1] and even deaths[2]. One way to assess whether autonomous machines are behaving in our best interest is to "stress test" (Falco & Gilpin, 2021) them with a set out-of-domain or tricky examples. These tests can consist of physical testing, e.g., putting the mechanism in unfavorable conditions like extreme temperatures, or algorithmic testing, e.g., constructing out-of-sample tests that are real-world corner cases or adversarial examples.

In the computer vision literature, these "stress tests" are commonly referred to as adversarial examples (Szegedy et al., 2013). These "tricks" are applied to images in the form of filters, noise, and color changes. There have been reports of these adversarial attacks in the real world (Eykholt et al., 2018), where just a few pieces of tape can change an object recognition model from perceiving a stop sign to a 45 mph sign. This naturally occurring adversarial example can be hard to find. Some other examples include traffic lights in a truck[3], in Figure 3b, or the moon being perceived as a yellow light[4] in Figure 3a. While these examples exist, aggregating a set of naturally occurring adversarial examples is difficult because naturally occurring adversarial examples are difficult to find and document.

Our work in anticipatory thinking assessment is inspired by similar work in AI assessment (Hernández-Orallo et al., 2017). Hernández-Orallo et al. argue that "to move ahead, the space of tasks must be analyzed." They note that this can be done by creating a hierarchy of tasks Hernández-Orallo (2017) or using task theory Thórisson et al. (2016). Our work defines such a task in terms of anticipatory thinking, and we show one type of task in the domain of object and image clasification.

2.3 Object/Image classification

Self-driving cars perceive the world in the modalities of images and sensors. Image processing is the most commonly used perception modality for self-driving cars[5] as it is inexpensive in hardware and software solutions, e.g., DNNs, can be pre-trained. Reliance on pre-trained models results in a need to manage risks, including unknown risks and observation biases.

Managing risks in self-driving is an open-world problem (Langley, 2020). The key issue is that many risks may be unknown, and therefore difficult to test. An especially pressing issue is how to design testing frameworks for such rare scenes. For example, a computer vision system

1. Mall robot injures a toddler: https://qz.com/730086/a-robot-mall-cop-did-more-harm-than-good/.
2. Uber self-driving car accident: https://www.wired.com/story/uber-self-driving-crash-arizona-ntsb-report/
3. https://futurism.com/the-byte/tesla-autopilot-bamboozled-truck-traffic-lights
4. https://www.ndtv.com/offbeat/watch-tesla-autopilot-feature-mistakes-moon-for-yellow-traffic-light-2495804
5. Tesla *exclusively* uses image processing for vehicle perception: https://venturebeat.com/business/tesla-ai-chief-explains-why-self-driving-cars-dont-need-lidar/

can be "fooled" by rare scenes such as traffic lights on a truck[6] (and see Figure 3b). In these cases, the object-level perception is correct: the objects are correctly identified, but the scene is so rare and disruptive that it should be immediately recognized so the system can avoid catastrophic failures (e.g. immediately stopping on a freeway). These "naturally-occurring adversarial scenes" exemplify the problem with the long tail of errors in autonomous vehicles. An additional layer of reasoning is required to test and monitor for these types of rare scenes.

Another issue in self-driving image models is observation bias. Pre-trained image models are prone to memorizing immeasurable patterns in training data Zhang et al. (2021). Adversarial robustness has developed as a subfield to combat such issues.

2.4 Adversarial Robustness

The current state-of-the-art vision systems are easily fooled (Nguyen et al., 2015) by out-of-distribution inputs that exploit a model's observational bias. Some solutions include training on adversarial examples (Ilyas et al., 2019), but they only lead to improvements on this data set. The adversarial robustness field is concerned with a set of modifications a malicious actor might execute with the intention of foiling a trained model. While this is certainly an important issue, it differs from our work in that a malicious actor strives to make imperceptible modifications (to the human eye) to model inputs (e.g. add noise to images) whereas our work modifies the semantics images (e.g. add weather effects). Adversarial robustness is designed to guarantee against known failure modes whereas our approach is designed to reveal hidden failure modes.

2.5 Synthetic data

Synthetic data, which could be generated from a GAN, VAE, or game engine, is an ideal compliment to our work. While generating synthetic data is not the focus of our work it serves as an important step in a revealing failure modes. Synthetic data generation is becoming more realistic and able to introduce semantics at a fine grain details using natural language, a key feature when directing generation from knowledge bases with human readable data (ie. knowledge graphs).

Many synthetic data sets exist for image processing Ros et al. (2016) and specifically for autonomous driving Li et al. (2019). Sensor data for autonomous driving can also be synthesized Yang et al. (2020). The key issue is that many of these datasets may not be realistic, especially when deployed and tested in simulation Reway et al. (2020). The key contribution of our work is to use real data, and slightly augment it to generate a synthetic sample sample. We use this synthetic input to force the existing model into a different class.

3. Anticipatory Thinking Alignment

Our goal is to define a procedure that assesses how an agent's performance in a set of tasks align with the anticipatory thinking used in different decision making frameworks. The key step that enables us to assess anticipatory thinking is synthesizing plausible, naturally occurring adversarial inputs that elicit different responses across the frameworks. While this method generalizes to other

6. https://futurism.com/the-byte/tesla-autopilot-bamboozled-truck-traffic-lights

domains, we use a running example with an autonomous vehicle's perception system. Driving is accessible to readers – we use stopping at stops signs as the task – and public pre-trained object identification models are highly accurate, reducing confounding factors. The goal of the running example is to illustrate how our contributions assess the anticipatory thinking alignment between AI systems and human decision making frameworks.

In our running example, we use an autonomous vehicle, AGENT, equipped with a trained object recognition system to perceive the road signs. We wish to assess how AGENT uses anticipatory thinking when driving and define the assessment as:

Definition 1 (Anticipatory Thinking Assessment) *A tuple* $Assess_{\text{AT}} = \langle T, D, KG, S, A \rangle$ *where* T *is a set of tasks,* D *the decision makers to align with,* KG *is a knowledge graph,* S *a set of synthesis functions,* A *the alignment function.*

Tasks are observable action(s) an agent takes in response to perceiving an object. Tasks prioritize all possible perceptions and actions an agent might make to those that manage risk – thus require anticipatory thinking– and are observable in the domain.

Definition 2 (Task) t_{obj} *is an observable agent task* t *when the agent perceives the object* obj*. The set of all anticipatory thinking tasks in an assessment is* T

In our driving example we use $stop_{ss}$ where the task $stop$ is an agent stopping at a stop line when perceiving a stop sign object, ss. Whether to obey the stop sign is a judgement of risk requiring anticipatory thinking. For illustrative purposes, we focus on this task but a short list of other driving related tasks and objects is in Table 1.

Anticipatory thinking varies widely across scenarios, as such we represent the variations as different decision makers. Each decision maker employs anticipatory thinking in a different way as a result of individual differences (e.g. domain expertise, temperament) or structured decision making frameworks (e.g risk aversion). We represent the different approaches to executing a task as:

Definition 3 (Decision makers) *is a set of decision making agents* D *where each agent* d_k *represents a unique decision making framework and* $d_k(t_{obj})$ *outputs a range of aligned values for task* t_{obj} *.*

In our driving domain, $DRIVERS$ consists of two driving styles; defensive $driver_def$ and aggressive $driver_agg$. In our stop at a stop sign task, $stop_{ss}$, our defensive driving agent would stop $95\% + -3$ and our aggressive agent stops $50\% + -5$. These values are not empirically determined (though they could be) and are for illustration only.

We expect human decision makers in $DRIVERS$ to behave consistently when observing slight variations of a typical stop sign. Whether this is red color fading from weather exposure, confounding shapes, occlusions, or otherwise their anticipatory thinking can account for the inferences needed to recognize and stop at a stop sign. We investigate $AGENT$'s anticipatory thinking ability by perturbing properties of stop signs to synthesize new ones, presenting them to $AGENT$, and identifying which decision maker $AGENT$ aligns with.

Object/Node	Task
Stop Sign	Stop at the stop line
Yield sign	Yield to moving objects (people, cars, animals)
Speed Limit Sign	Velocity magnitude must be $\leq$ the speed limit
No parking	Do not park car at sidewalk
Flashing red or yellow lights	Stop and yield to moving objects.

Table 1: Examples of tasks that have risk and their observable behavior.

Procedure 1 An anticipatory thinking ability assessment procedure that calculates an agent's alignment with decision making frameworks.

Given: an assessment $Assess_{\text{AT}} = \langle T, D, KG, S, A \rangle$, corpus of object instances C, and an agent a.

Find: a set D_a^C whose elements correspond to how aligned a is to each decision maker.

procedure ANTICIPATORY-THINKING-ASSESSMENT($Assess_{\text{AT}}$, C, a)
  Let $R \leftarrow \varnothing$ ▷ Start with an initially empty results set.
  for each task-object $t_i \in T \in Assess_{\text{AT}}$ **do** ▷ for each task-object in the assessment
    for each property $p_j \in i(KG)$ **do** ▷ for each property of object i in knowledge graph KG
      Let $S_{ij} \leftarrow f_p(i(C))$ ▷ synthesize new instances of object i by modifying property p_j.
      Let $R_{ij} \leftarrow t(S_{ij})$AGENT ▷ Results R from a doing task t_i with new instances.
  Let $D_a^C \leftarrow A(R, D, a)$ ▷ Align an agent to decision maker responses
  return D_a^C

Our approach to synthesizing perceptions relies on generating plausible variations on real-world objects in the domain. We ensure our synthesized perceptions are aligned with real-world knowledge by using a knowledge graph that represents the commonly agreed upon relationships between objects and their properties. We use the typical formulation of a knowledge graph

$$KG = (V, E)$$

as collection of linked properties of vertices, V, via relational edges, E. In our definition of a knowledge graph, the vertices, which we will refer to from now on as *nodes* are objects, e.g., stop sign, vehicle, or penguin. We represent a datum in the knowledge graph as a triple: `(query, relation, object)`, where the query and object are nodes. Nodes are connected to other nodes via edges which encode relationship properties, e.g., `IsA`, `HasA`, `AtLocation`, etc.

When an autonomous vehicle arrives at a stop sign, which has been perceived by the vision system. Stop signs have higher level properties, e.g., shape: hexagon and color: red, that prompt an expected behavior: the vehicle must stop. Each of these higher level properties can be interrogated for variations that affect an agent's ability to perform the tasks. To verify that an autonomous vehicle's underlying perception system is is safe, secure, or trustworthy, we need to *identify* the nodes in our tasks, and *validate* or *stress test* that these properties are met.

Our AT assessments are designed to elicit hidden failure modes that might otherwise be missed. The kernel of each assessment is a corpus of real-world instances of the objects that initiate tasks in the assessment:

Definition 4 (Corpus) *is a set C of tuples $\langle i, l \rangle$ where i is object instance and l the label from the objects needed to initiate the tasks in T. $l(C)$ returns all instances with label l in the corpus.*

To elicit hidden failures in the $stop_{ss}$ task, we collect images of stop signs $ss_1, ..., ss_2$ into our corpus DB. This would typically be the data that a machine learning algorithm would train on.

For all object labels DB, we identify them in a knowledge graph and enumerate through their properties to identify those for which we have a synthesis function. The intention with a synthesis function is to generate the naturally occurring variations that are plausible in a given context. We define a synthesis function as:

Definition 5 (Synthesis function) *is a function $f_p(n(C))$ that modifies property p of the instances of object n in the corpus C and outputs the modified instances in a set n'_p.*

Continuing with our stop sign example, the stop sign node ss has a property $color = red$ and we have a synthesis function $f^{red}(ss)$ that changes the ss's red channel value in the output ss'_{red}. This function is a simplified way to represent the fading and weathering stop signs endure from exposure the elements.

Lastly, we define an alignment function to assess an agent's task performance on all synthesized data against those of our decision makers. Alignment functions can take many forms including trained classifiers in data rich environments, to few-shot learning when novelty is a factor to rule-based systems in well-defined domains.

Definition 6 (Alignment) *is a function $align(C, Assess_{AT}, a)$ that outputs D^C_A as agent a's alignment with the decision makers and tasks in the assessment $assess_{AT}$ on the corpus C.*

Our alignment function is a T-test that tests whether $AGENT$'s performance on the synthesized data is drawn from the same distribution as our aggressive and defensive drivers. We find that $t\text{-}test(DB', Assess_{AT}, \text{AGENT})$ returns $D_{\text{A}}^{DB'} = \{0, 1\}$ indicating that $AGENT$ aligns with a more aggressive driver. One can envision more advanced methods to determine alignment from topological data analysis if there are many tasks and synthesis functions.

4. Evaluation

Our goal is characterize model robustness by (1) stress-testing image perturbations and (2) characterizing model robustness with a given knowledge graph. We stress-test image perturbations by adding an image mask for each color channel. We apply this image mask for each color channel, and iteratively; for various levels of modifications (up to 30 iterations). This same approach could be used for other symbolic features: background blurring, removing objects, etc. In our initial evaluation, we focus on color channels, which aligns with the top component of the knowledge graph.

We are simplifying the decisions that an autonomous vehicle would make by focusing on object detection and color channels. There can be multiple inputs that are tasked with perceiving the world: image/object recognition, LiDAR, radar, etc. However, many autonomous vehicle manufacturers have decided to fully focus on image/object detection[7]. In this work, we focus on recognizing a stop sign because it is a necessary condition to stop at a stop sign. Of course, there are other tasks to recognize when to stop. However, the stop sign is an almost universal symbol across locals and countries to stop. The other problem is that stop signs can also be carried by pedestrians that indicate stopping in critical situations like road work or children crossing.

We characterize the model robustness by querying ConceptNet5 (Speer et al., 2017) for key semantic parts of the image. We use a domain-specific knowledge graph to aggregate semantic knowledge about objects and their properties. We use ConceptNet5 Speer et al. (2017), a semantic knowledge base of commonsense that is able to query for concepts that contain multiple words, e.g., stop sign. An example of the output of querying ConceptNet for "stop sign" are in Table 3. This shows that color (the second line in the table) are a key factor for identifying a stop sign.

We aggregated a dataset for the AT assessment fromfrom Kaggle[8]. We pruned the set of images that were identified as stop signs for a total of 91 stop signs. We used an existing pre-trained model, ResNet50, and put into evaluation mode, which is necessary to e.g. turn off batchnorm. ResNet50 was not trained on the "stop sign" label, so we use the "street sign" label. From the 91 initial stop signs, we choose 10 stop sign images to test. One of the images contained both a stop sign and a traffic light so it was characterized as a traffic light (index 67) in Table 2. The other 9 images were chosen due to several factors:

1. True label: The predicted label and the true label are both "street sign" or "stop sign."
2. Lack of background noise: The image's background is simple. Since we are modifying the image at the pixel level, if the image has a complicated or busy background, our approach can fail.
3. Simple color scheme: The image's color scheme is simple. Since we are modifying the image at the pixel level, if the image contains many diverse colors, our approach can fail.
4. Focus on the sign: The main component of the image should be a stop sign. If the sign is in the background, it is usually not labelled correctly.

Our 10 testing images were specifically chosen with these four factors in mind. In future work, we will extend this work to more diverse backgrounds. For each image in our data set, we read the image, resize to 224 and convert to PyTorch Tensor. We then normalize each image's RGB values, and predict unmodified image tensor class. For all of our data set, the predicted unmodified image tensor class is "street sign." We then get the mask from the image that we will modify. We do this for each color channel: red, green and blue. And finally, we apply gradient descent in the following four steps:

1. Modify a single pixel's RGB values

7. Tesla does not use LiDAR data: https://towardsdatascience.com/why-tesla-wont-use-lidar-57c325ae2ed5
8. Kaggle Road Sign Detection Dataset: https://www.kaggle.com/datasets/andrewmvd/road-sign-detection

2. Color the image mask
3. Add the mask to the original image
4. Predict result (and check whether the prediction is different from the original prediction).

We report the first incorrectly classified images in Figure 4. From the figure we see that most images are incorrectly classified by the third or fourth iteration. The detailed results in Tables 2 and visualized in Figure 4. These results demonstrate that a state-of-the-art image classifier like ResNet50 is not very robust to small color changes. This is inspired by augmented stop signs that may be perceived in the real world, e.g., stop signs that are faded or contain stickers or graffiti. An example output is shown below for three iterations:

```
Iteration 1: Predicted class:  street_sign
Iteration 2: Predicted class:  street_sign
Iteration 3: Predicted class:  spotlight
```

This shows that on the third iteration, the top predicted class changed from street sign to spotlight. We continue this process for up to 30 iterations, and we report the final prediction in Table 2. It is important to note that 7/10 of the images are incorrectly classified for all color channels after 30 iterations.

Index	True label	Iter. B	B end label	Iter. R	R end label	Iter. G	G End Label
44	street sign	3	traffic light	3	spotlight	3	stoplight
50	street sign	3	coil	3	analog clock	3	coil
51	street sign	None	street sign	4	spotlight	None	street sign
58	street sign	17	traffic light	6	digital clock	4	traffic light
60	street sign	None	street sign	3	digital clock	23	bolo tie
63	street sign	3	spotlight	3	spotlight	3	stoplight
67	traffic light	3	digital clock	4	spotlight	8	stoplight
68	street sign	None	street sign	3	digital clock	3	stoplight
75	street sign	3	digital clock	4	spotlight	3	digital clock
78	street sign	3	spotlight	3	spotlight	3	stoplight

Table 2: A table showing the the true label, and the number of iterations to change the label for each color channel (Red=R, Blue=B, Green=G). We also show the end label after 30 iterations.

5. Limitations and Future Work

Our anticipatory thinking assessment approach can generalize to other applications. Here we should autonomous machine perception, particularly autonomous vehicle perception as a use case. In our initial evaluation, we focused on a stop sign and a single perception model, ResNet50. In future work, we will extend this assessment to multiple perception models and object types.

subject	predicate	object
a stop sign	AtLocation	the corner of two streets
a stop sign	HasProperty	red and white
a stop sign	UsedFor	stopping
a stop sign	AtLocation	a street corner
a stop sign	AtLocation	a fork in the road
stop sign	Synonym	stopbord
stop sign	RelatedTo	driver
stop signs	FormOf	stop sign
stop sign	IsA	road sign
a stop sign	UsedFor	controlling traffic
stop	Synonym	stop sign

Table 3: A subset of the top results for querying "stop sign" in ConceptNet. The colors are a top property of being considered a stop sign. Results were pruned that contained either contained (1) non-english results or (2) non-utf8 characters.

(a) The baseline image.

(b) An example of a mask for the green color channel (green eliminates red).

Figure 3: The mask that we apply for searching on different channels.

Our assessment is also limited for the sake of example. We use 30 iterations of the approach to demonstrate that most images are incorrectly labelled in the first 3-4 iterations. In future work, we may extend the number of iterations. We will also work on creating masks for different types of attacks. Currently, we focus on color as the main assessment, but we can extend this for different textures, object classes, and other properties identified in the knowledge graph.

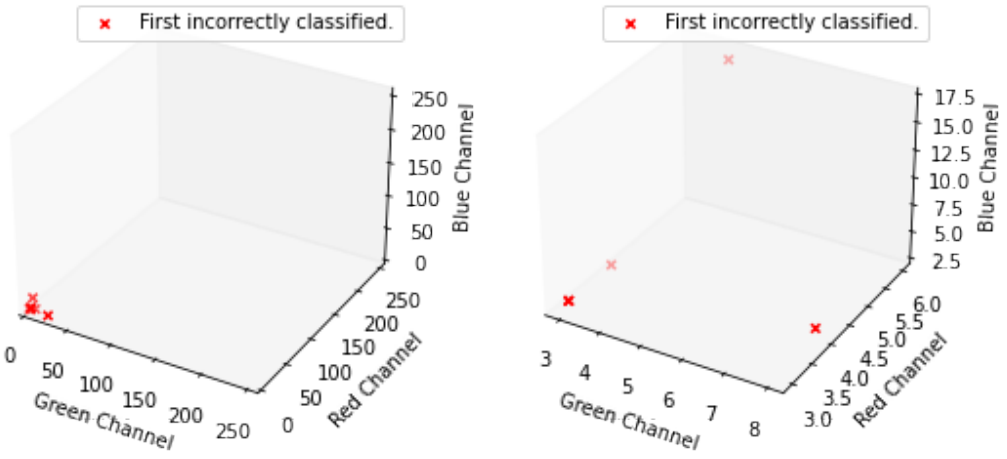

Figure 4: The 7/10 images that were incorrectly classified after 30 iterations viewed for all pixels (up to 256) and focused on the misclassifiation. We show the gradient descent steps to misclassification on all three color channels: red, green, and blue. The red x indicates the first incorrectly classified image, where a mask is added to the original image.

6. Discussion: On the AV Use Case

Autonomous systems have started making human-level decisions in safety-critical domains, mission-critical domains like medical diagnosis and autonomous driving. While autonomous vehicles (AVs) promise to be more efficient, economical, and even safe, they still make critical errors. These autonomous agents are increasingly deployed in real world settings, where there has been an increase in malfunctions and errors leading to injuries [9] and even deaths Marshall & Davies (2018). One way to assess whether autonomous machines are behaving in our best interest is to have them *explain* their behavior, to ensure that the underlying reasoning behind their decisions is sound and robust.

Many autonomous vehicle failures are due to errors in perception and lack of commonsense knowledge and reasoning. When a Tesla autonomous driving system confused the red text on a sign for a stop-light[10], the vision processing system failed because it had never seen that error before in *training*. This is a common failure case in perception systems, but it becomes critical when these same technologies are deployed on a self-driving car.

Autonomous vehicles are constantly perceiving their surrounding environments with perception systems, e.g., vision, LiDAR, and radar, to resolve the difference between reality and their representation of it. The vision system, which is opaque to humans, is not prepared for rare but highly impactful perception errors; making autonomous vehicles an open-world domain (Langley, 2020).

Autonomous vehicles have a long tail of errors, especially for the vehicle perception system. One of the challenges of quantifying the risk of autonomous vehicles in the real world is that they are

9. https://www.washingtonpost.com/technology/2022/06/15/tesla-autopilot-crashes/
10. Tesla Confuses Red Text on a Flag for a Stop Light https://www.thedrive.com/tech/37250/tesla-autopilot-confuses-red-text-on-a-flag-for-stop-light.

deployed in an open-world but tested in closed-world simulations. This juxtaposition highlights how unprepared autonomous vehicles are for naturally occurring adversarial examples. In this challenge, we suggest an alternative: generating naturally occurring adversarial examples from "real" rare cases to supplement the closed-world test simulations, mimicking the open-worlds where they are deployed.

The current state-of-the-art vision systems are easily fooled Nguyen et al. (2015) by out-of-distribution inputs that exploit a model's observational bias. Some solutions include training on adversarial examples Ilyas et al. (2019), but they only lead to improvements on this data set. Instead, we propose iterative testing where objects mislabelled in deployment are used as input to a content generation model for testing and evaluation of a perception system's observational biases. In application, generative model creates sets of naturally adversarial objects that can be tested during development (e.g. faded stop signs, shadow patterns).

7. Conclusion

Deploying AI systems in mission critical contexts or where public safety is concerned requires assurances that the system can avoid preventable failures, even in scenarios that are not explicitly evaluated. Humans operate safely across a variety of tasks in these contexts using anticipatory thinking, enabling them to manage emergent risks in new but not entirely novel scenarios. Assessing AI systems anticipatory thinking ability will characterize its ability to perform in the new scenarios.

Our approach assesses an agent's anticipatory thinking ability by generating new plausible data and aligning the agent's responses with known decision making frameworks. Our method generalizes beyond our autonomous vehicle and computer vision example because it is driven by modifying the semantics of the test inputs. Assessing AI systems based on their abilities – in this case anticipatory thinking – is new ground for the field and necessary to balance increased adoption. We demonstrated the value of generating synthetic data

In future work, we can extend our simple example by incorporating state of the art generation methods, such as GANs, to manipulate images at the scene level. For example, introducing a stop sign in a mural on the side of a building and assess AI system responses. Additionally, we will expand the assessment tasks and empirically evaluate them with respect to driving decision frameworks using human subjects. This will improve the alignment task and add comparative value of increasing the number of image systems we can assess beyond the single pre-trained model in our example.

Acknowledgements

Please place your acknowledgements in an unnumbered section at the end of the paper. Typically, this will include thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies or corporate sponsors that provided financial support.

References

Amos-Binks, A., & Dannenhauer, D. (2019). Anticipatory thinking: A metacognitive capability. *arXiv preprint arXiv:1906.12249*.

Amos-Binks, A., Dannenhauer, D., & Gilpin, L. H. (2021). Anticipatory thinking challenges in open worlds: Risk management.

Davis, E., & Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, *58*, 92–103. From https://doi.org/10.1145/2701413.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., & Song, D. (2018). Robust physical-world attacks on deep learning visual classification. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1625–1634).

Falco, G., & Gilpin, L. H. (2021). A stress testing framework for autonomous system verification and validation (v&v). *2021 IEEE International Conference on Autonomous Systems (ICAS)* (pp. 1–5). IEEE.

Geden, M., Smith, A., Campbell, J., Spain, R., Amos-Binks, A., Mott, B., Feng, J., & Lester, J. (2019). Construction and Validation of an Anticipatory Thinking Assessment. *Frontiers in Psychology*, *10*, 1–10.

Gilpin, L. H., Macbeth, J. C., & Florentine, E. (2018). Monitoring scene understanders with conceptual primitive decomposition and commonsense knowledge. *Advances in Cognitive Systems*, *6*, 45–63.

Hernández-Orallo, J. (2017). Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. *Artificial Intelligence Review*, *48*, 397–447.

Hernández-Orallo, J., Baroni, M., Bieger, J., Chmait, N., Dowe, D. L., Hofmann, K., Martínez-Plumed, F., Strannegård, C., & Thórisson, K. R. (2017). A new ai evaluation cosmos: Ready to play the game? *AI Magazine*, *38*, 66–69.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*.

Langley, P. (2020). Open-world learning for radically autonomous agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*, 13539–13543. From https://ojs.aaai.org/index.php/AAAI/article/view/7078.

Li, X., Wang, Y., Yan, L., Wang, K., Deng, F., & Wang, F.-Y. (2019). Paralleleye-cs: A new dataset of synthetic images for testing the visual intelligence of intelligent vehicles. *IEEE Transactions on Vehicular Technology*, *68*, 9619–9631.

Marshall, A., & Davies, A. (2018). Uber's Self-Driving Car Saw the Woman It Killed, Report Says. https://www.wired.com/story/uber-self-driving-crash-arizona-ntsb-report/.

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 427–436).

Reway, F., Hoffmann, A., Wachtel, D., Huber, W., Knoll, A., & Ribeiro, E. (2020). Test method for measuring the simulation-to-reality gap of camera-based object detection algorithms for autonomous driving. *2020 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1249–1256). IEEE.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. *Proceedings of*

the IEEE conference on computer vision and pattern recognition (pp. 3234–3243).

Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. *Thirty-first AAAI conference on artificial intelligence*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Thórisson, K. R., Bieger, J., Thorarensen, T., Sigurðardóttir, J. S., & Steunebrink, B. R. (2016). Why artificial intelligence needs a task theory. *International Conference on Artificial General Intelligence* (pp. 118–128). Springer.

Yang, Z., Chai, Y., Anguelov, D., Zhou, Y., Sun, P., Erhan, D., Rafferty, S., & Kretzschmar, H. (2020). Surfelgan: Synthesizing realistic sensor data for autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11118–11127).

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*, 107–115.